Z80 Program Technique

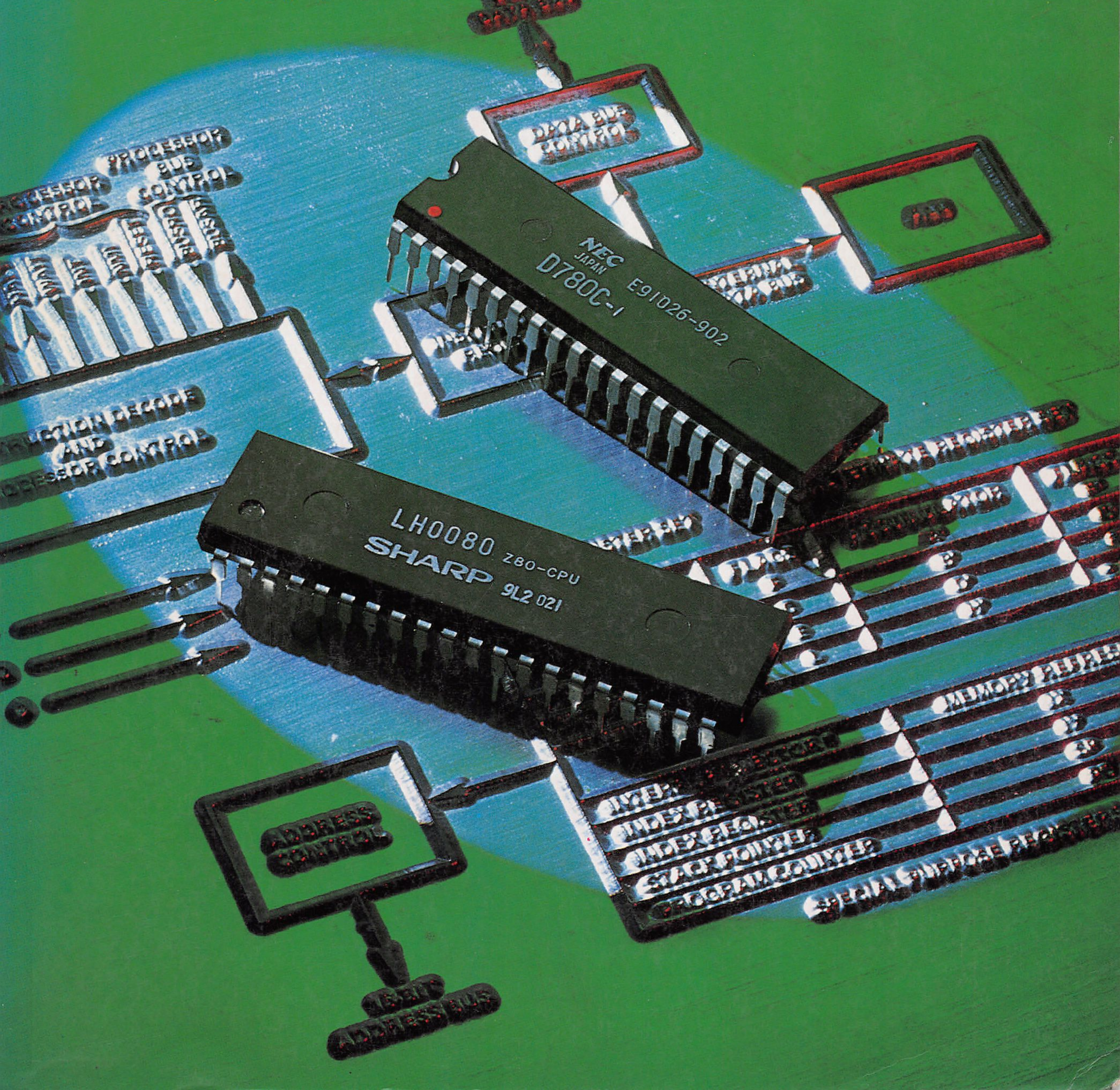
月刊マイコン別冊

Z80マイコン

プログラムテクニック

■Z80命令群をわかりやすく解説

■テストプログラム、演算プログラムパッケージ付



NECのコンピュータ&LSI技術が生んだ

Personal Computer PC-8000 Series

たしかな技術で世界をむすぶ NEC

キャストは決まった!!



周辺機器も増々充実。
パーソナルコンピュータ革命をリードするPC-8000シリーズ。

- スタンドアローンマシンから端末まで幅広い用途に応える、パーソナルコンピュータPC-8000シリーズ。
- さまざまなアプリケーションを支える豊富な周辺機器が勢揃い。



型名	品名	価格
PC-8001	本体(パーソナルコンピュータ)	168,000円
PC-8005	増設メモリパック	24,500円
PC-8011	拡張ユニット	148,000円
PC-8021	80桁 ドットインパクトプリンタ	165,000円
PC-8022	40桁 サーマルプリンタ	98,000円
PC-8031	デュアルミニディスク・ユニット	310,000円
PC-8032	拡張用デュアルミニディスク・ユニット	
PC-8041	12インチ・グリーンディスプレイ	48,800円
PC-8042	12インチ・カラーディスプレイ	109,000円
PC-8043	12インチ・カラー(高解像度)ディスプレイ	219,000円
PC-8033	PC-8031用 I/Oポート	17,000円
PC-8044	家庭テレビ用カラーアダプタ	13,500円
PC-8062	RS-232Cケーブル	18,700円

Bit-INN TOKYO
〒101 東京都千代田区外神田1-15-16
ラジオ会館7F ☎(03)255-4575～6
●東日本地区通信販売店
日本電子販売株式会社
☎(03)255-4571(代)

Bit-INN OSAKA
〒542 大阪市南区難波新地6番町10-1
マスザキヤビル4・5F ☎(06)647-2747～8
●西日本地区通信販売店
ミカサ商事株式会社
☎(06)942-1941(代)

Bit-INN NAGOYA
〒460 名古屋市中区大須4-11-5
杏林殖産ビル2F ☎(052)263-0971
●中部地区通信販売店
萩原電気株式会社
☎(052)931-3511(代)

Bit-INN YOKOHAMA
〒220 横浜市西区北幸1-8-4
横浜西口第2ミナトビル7F ☎(045)314-7707～9
●通信販売店
日本マイクロコンピュータ株式会社
☎(03)230-0041(代)

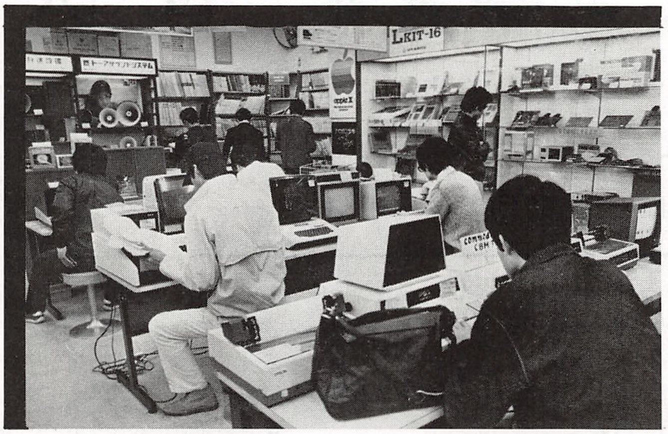
日本電気株式会社
本社 〒108 東京都港区芝五丁目33-1〔日本電気本社ビル〕☎(03)454-1111(大代)
電子デバイス販売事業部マイクロコンピュータ販売部 〒108 東京都港区芝五丁目33-7〔徳栄ビル〕☎(03)453-5511(大代)

大阪・日本橋 マイコンショップ

東亜エレシャック toa

1階 マイクロコンピュータ専門コーナー

2階 アマチュア無線機器と電子機器 オーディオキットコーナー



一日中座り続けている熱心なファンもたくさんいます



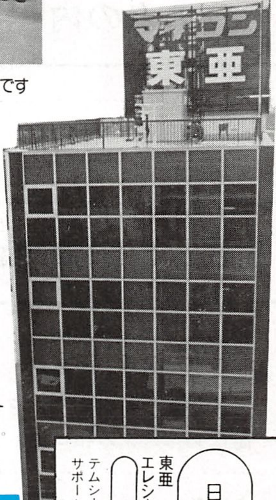
話題のTRS-80コーナーもフルシステムを展示、充実しています



広い店内には豊富な商品群、目玉商品などが数多く置かれています



皆んな真剣な目付きです



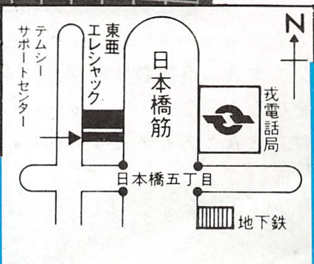
《取扱いメーカー》

Tandy Radio Shack　commodore　apple II　NEC
HITACHI　SHARP　TOSHIBA
pf パナファコム株式会社　EPSON　テキサス インスツルメンツ アジア リミテッド
tomcy 東亜マイクロコンピュータ

関連週辺機器・ソフト関係・専門書籍

※お手持ちの不要マイコン(システム・1ボード型)を下取・委託販売いたします。詳細は係員までご相談ください。
※ローン、クレジット及び通信販売も取扱っています。(10,000円以上の通信販売は、運賃サービスいたします。)

toa 東亜無線グループ
東亜エレシャック株式会社
〒556大阪市浪速区日本橋筋5-61 TEL06(644)0111(代)
地下鉄堺筋線恵美須町北出口右前
営業時間 AM10:00~PM6:30 定休日 毎週木曜日



Z80マイコン プログラム・テクニック—目 次—

目次

目　次

月刊マイコン 別冊

Z80マイコンプログラムテクニック

（庄司 渉，石田 稔 著）

昭和55年4月30日発行

定価1,300円（送料200円）

編集発行人　平 山 秀 雄
発 行 所　(株)電波新聞社
郵便番号141 東京都品川区東五反田1－11－15
電話(03) 445－6111(大代) 振替東京5－51961
印 刷 所　大日本印刷(株)
製 本 所　(株)堅 省 堂

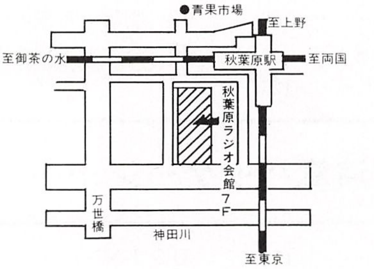
NEC Bit-INN

初心者からベテランの方までのマイコンに関する
あらゆる問題を解決するサービスルームです。
皆様のはば広いご利用をお待ちしております。

Bit-INN 東京

〒101 東京都千代田区外神田1−15−16
ラジオ会館７F ☎(03)255−4575〜6

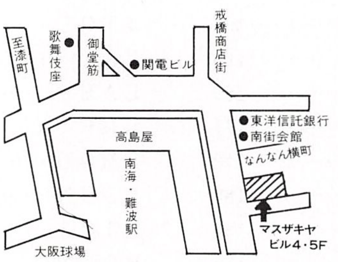
●東日本地区通信販売店
日本電子販売株式会社
〒101 東京都千代田区外神田1−16−1
万世ビル３F ☎(03)255−4571㈹



Bit-INN 大阪

〒542 大阪市南区難波新地６番町10−1
マスザキヤビル4・5F ☎(06)647−2747〜8

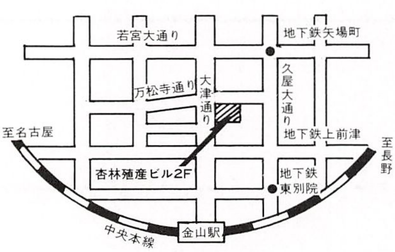
●西日本地区通信販売店
ミカサ商事株式会社
〒540 大阪市東区島町2−5
☎(06)942−1941㈹



Bit-INN 名古屋

〒460 名古屋市中区大須4−11−5
杏林殖産ビル２F ☎(052)263−0971
（地下鉄、上前津駅下車、万松寺方面へ）

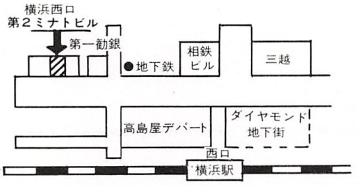
●中部地区通信販売店
萩原電気株式会社
〒461 名古屋市東区東桜2−3−3
☎(052)931−3511



Bit-INN 横浜

〒220 横浜市西区北幸1−8−4
横浜西口第２ミナトビル７F
☎(045)314−7707〜9

●通信販売店
日本マイクロコンピュータ株式会社
〒102 東京都千代田区麹町4-5-21睦ビル７F
☎(03)230−0041㈹



●入門コーナー
マイコンをはじめて覚えようとする方のために簡単なデモセットを用意し、ご自由にご利用いただくコーナーです。

●マシンコーナー
ＰＤＡ−80をはじめプリンタ等を用意してあり、ＲＯＭ書き込み、デバック、プログラムリストの製作にご利用下さい。

●技術相談コーナー
マイクロコンピュータに関する、あらゆる問題の解決に専門技術者が常時待機し、各産業分野、専門分野へのマイクロコンピュータの応用、導入へのためのアドバイザーとして、技術相談に応じております。

●展示コーナー
パーソナルコンピュータＰＣ−8001を中心に、マイコンセットを準備し、ご自由にお使いいただいております。ご自分の応用プログラミングにもご利用ください。

●資料コーナー
マイコンに関する、初歩から専門分野までの資料を用意し、すべての資料の販売を行っており、資料の豊富さでは業界一とご好評を得ております。

●販売コーナー
マイコンとその周辺機器類及び関連LSI、工具類等を豊富に取揃え、地方の方のためには通信販売も行っております。尚便利でお買い求めやすいＮＥＣクレジットもございますのでご利用下さい。

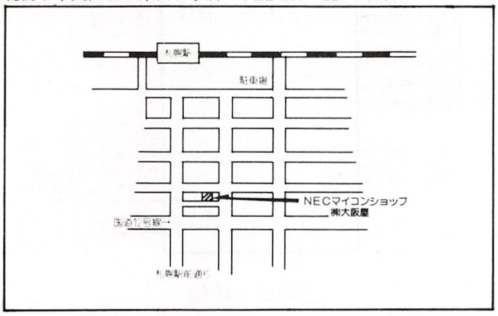
●修理コーナー
各種の工具、測定機器類を用意し、ベテランの技術者がご相談に応じております。マイコンの故障でお困りの方や自分で「手作りのマイコン」を作りたい方などのご利用をお待ちしております。

NEC ぞくぞくOPEN!

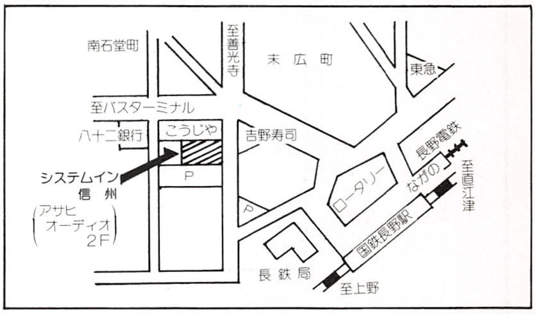
マイコンショップ

あなたのパートナー、マイコンショップ
のニューフェイス、各地区に開店!
Bit-INN同様お気軽にご利用下さい。

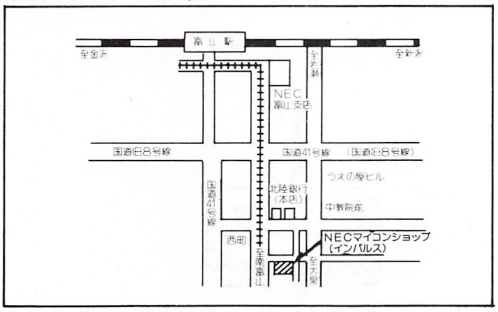
● (北海道地区) ㈱大阪屋
札幌市中央区北1条西3丁目　TEL 011―221―0181



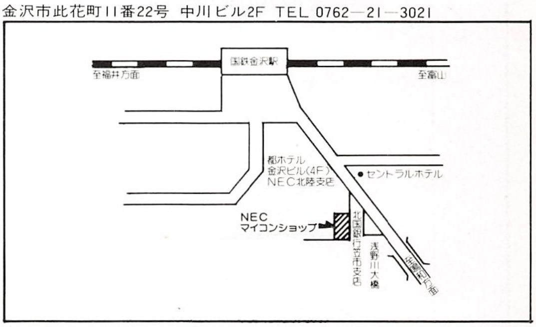
● (長野地区) システムイン信州
長野市南石堂町1282 アサヒオーディオ2F TEL 0262―27―6136



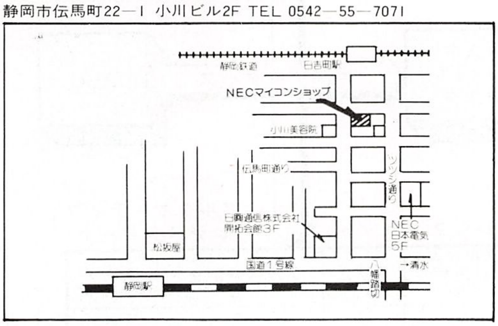
● (富山地区) インパルス
富山市五番町4―10 西野ビル2F　TEL 0764―91―2212



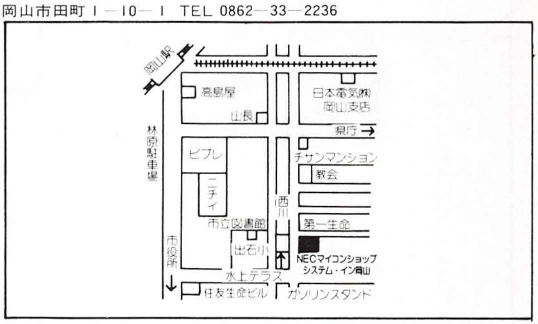
● (石川地区) 北陸マイクロコンピュータ販売
金沢市此花町11番22号　中川ビル2F　TEL 0762―21―3021



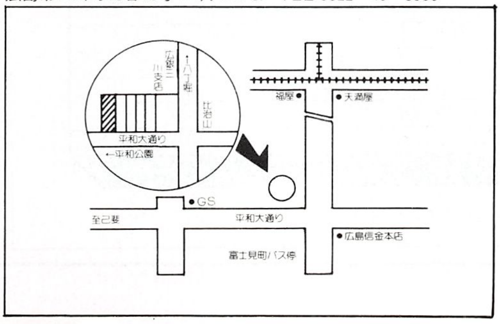
● (静岡地区) 日興通信静岡支店
静岡市伝馬町22―1 小川ビル2F TEL 0542―55―7071



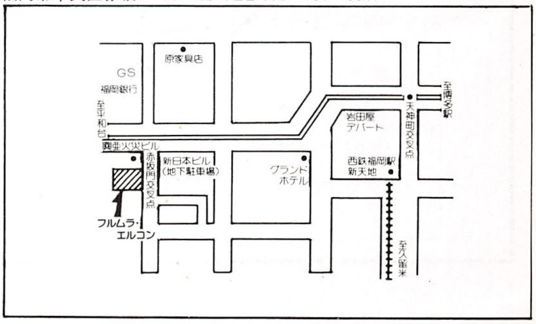
● (岡山地区) システム―イン岡山
岡山市田町1―10―1 TEL 0862―33―2236

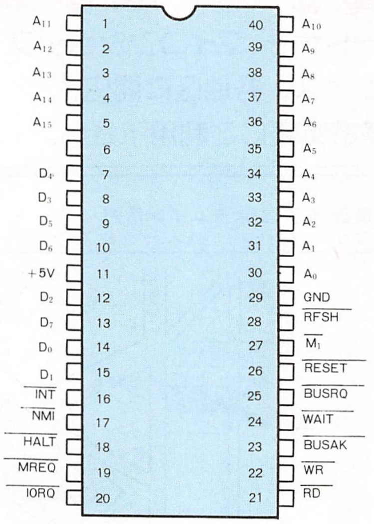


● (広島地区) インタフェース
広島市三川町10番10号 三角ビル3F TEL 0822―49―3950

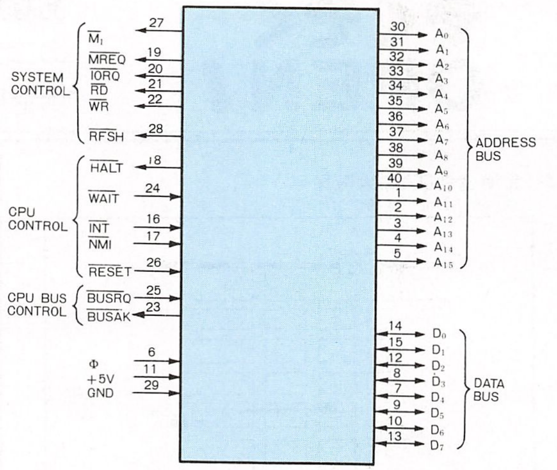


● (福岡地区) フルムラ・エルコン
福岡市中央区赤坂1―10―22 TEL 092―751―6647



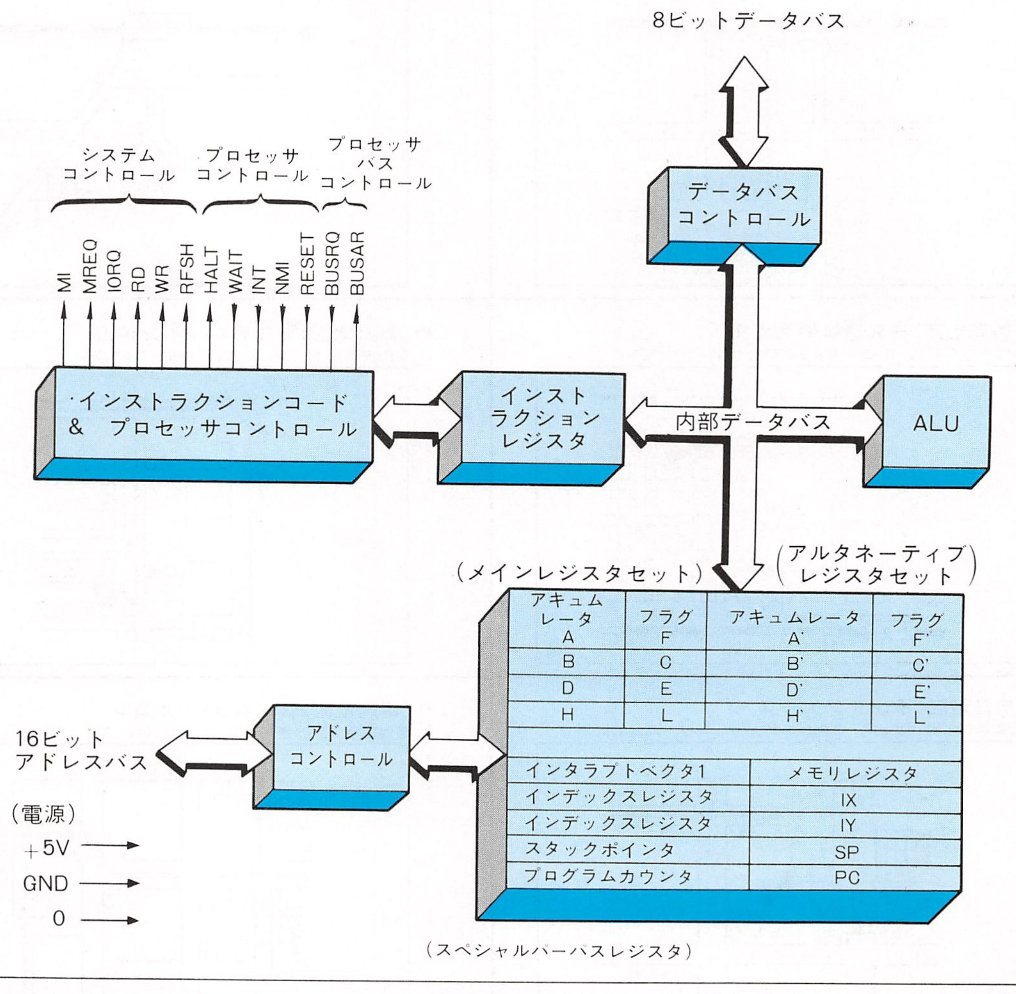


Z80外観(上面より)



Z80ピン接続

Z80内部構成



Z80とは

インテル社の開発したCPU8080が，その手軽さと汎用性のために，8ビットマイクロコンピュータの主流になったことは，すでに御存知の通りです。

こういった世界では，一度主流になると普及の速さには，さらに拍車がかかるものです。そしてこの8080用のソフトウエアの蓄積量は，膨大なものになりました。

時がたち，多くの人が8080に慣れ，すべてを知るようになると，その限界がより明確な形で見えてくるようになりました。一方，ひとたび普及し始めたマイクロコンピュータに対する各方面からの期待も日に日に大きくなっていきます。

かといって，まったく新しいCPUを開発したのでは，すでにある膨大な量のソフトウエアをすべて使いものにならないものにしてしまいます。

そこで，命令体系が複雑になるのは覚悟の上で，8080の全命令が機械語レベルで使え，さらに8080では不十分だった機能を持たせる命令を追加するという方向で，開発されたのが，ザイログ社のZ80というわけです。

では，いったい，8080のどこに限界があり，そのためにZ80ではどんな機能が追加されたのか，お話しすることにしましょう。

1. レジスタの数がふえたこと

8080は，

A ：アキュームレータ　（8ビット）
B ：レジスタB　（8ビット）
C ：レジスタC　（8ビット）
D ：レジスタD　（8ビット）
E ：レジスタE　（8ビット）
H ：レジスタH　（8ビット）
L ：レジスタL　（8ビット）
F ：フラグレジスタ　（8ビット）
SP：スタックポインタ　（16ビット）
PC：プログラムカウンタ（16ビット）

というレジスタを持っていますが，Z80は，これらのレジスタに加えて，次のようなレジスタを持っています。

A′ ：アキュームレータA′（8ビット）
B′ ：レジスタB′　（8ビット）
C′ ：レジスタC′　（8ビット）
D′ ：レジスタD′　（8ビット）
E′ ：レジスタE′　（8ビット）
F′ ：フラグレジスタF′（8ビット）
IX ：インデックスレジスタIX（16ビット）
IY ：インデックスレジスタIY（16ビット）
I ：インタラプトベクタレジスタI（8ビット）
R ：メモリリフレッシュレジスタR（8ビット）

レジスタA′～F′は，各々レジスタA～Fの裏レジスタにあたり，したがって，たとえばAとA′を同時に使うことはできません。

レジスタIX，IYは，8080では不可能であった相対アドレス指定，番地修飾の命令追加にともなって必要となってきたレジスタです。

レジスタIは，割り込み機能拡張にともなって

必要になってきたレジスタです。

レジスタRは，8080では直接つなげることができなかったダイナミックラムをつなげるようにしたために，必要となってきたレジスタです。

2.相対アドレス指定が可能になった

8080の各種ジャンプ命令で，そのジャンプ先を指定する場合，００００からＦＦＦＦまでの絶対アドレスを用いて指定していました。たとえ，五つ先のアドレスへジャンプする場合でも，16進数4桁の数でそのアドレスを指定しなければいけませんでした。これはメモリの効率良い使い方とは，決していえません。

さらに，このようにジャンプ先を指定する場合，命令を書き込むアドレスを後になって変更したいときには大変です。すべてのジャンプ命令をひろい出し，そのジャンプ先を変えてやらなければなりません。

そういったときに便利なのが相対アドレス方式で，この命令より何番地先のアドレスへジャンプとか，何番地後のアドレスへジャンプという形でジャンプ先のアドレスを指定するやり方です。

Z80には，－126から＋129の範囲でこの相対アドレス指定ができるジャンプ命令が，６個用意されています。ジャンプ先のシンボルをＪＭＰとしたとき，その各命令をアセンブラ言語で表わしてみると，

ＪＲ　ＪＭＰ	：無条件ジャンプ
ＪＲ　Ｃ，ＪＭＰ	：CY＝１ならジャンプ
ＪＲ　ＮＣ，ＪＭＰ	：CY＝０ならジャンプ
ＪＲ　Ｚ，ＪＭＰ	：Ｚ＝１ならジャンプ
ＪＲ　ＮＺ，ＪＭＰ	：Ｚ＝０ならジャンプ
ＤＪＮＺ　ＪＭＰ	：Ｂ←Ｂ－１実行後，Ｂ≠０ならジャンプ

これらの命令はすべて２ワード命令なので，8080のジャンプ命令(これはすべて３ワード命令)より，１ワード少なくてすみます。

3.対象アドレスの指定方法が豊富になった。

レジスタの数がふえた，というところでもお話ししましたが，Z80は二つのインデックスレジスタＩＸ，ＩＹを持っています。そのため，対象アドレスの指定方法が豊富になっています。

その使い方ですが，8080でＨＬレジスタペアを用いて対象アドレスを指定していた部分を

IX＋d, IY＋d(ただしdは正負を考慮した８ビット以内の数）で置きかえることができる，と考えればよいでしょう。

たとえば，

ADD A，(IX＋d)

は，アキュームレータＡの内容と，インデックスレジスタＩＸの内容に８ビットの数ｄを加わえたときにできる数をアドレスとするメモリの内容とを加わえ，その結果をアキュームレータＡに入れる，という命令です。

ただし，ＨＬレジスタペアの代用として用いる場合には，２ワード余分にメモリを使うことになることを覚悟しなければなりません。

4.16ビットの数が扱いやすくなった

インデックスレジスタIX, IYがあることもあって，16ビットの数をそのまま，ロードしたり，演算したりする命令がふえています。

たとえば

LD BC，(DATA)
　　………BC←M(DATA)
ADC HL，DE
　　………HL←HL＋DE＋CY
SBC HL，BC
　　………HL←HL－BC－CY

などは，たいへん便利です。

（注）ＣＹはキャリーのことです。

5.回転(ROTATE)命令が強化され，シフト命令が追加されたこと

8080のROTATE命令は，アキュームレータＡを介してでなければできませんでした。Z80では，どのレジスタでも，それが可能になっています。もちろんＨＬレジスタペアインデックスレジスタIX，IYで指定されるメモリの内容のROTATEも可能です。

また，回転（ＲＯＴＡＴＥ）ではなく，シフト

(SHIFT)命令が追加され，2の累乗の計算が，手際よく実行できるようになっています。

その他，4ビット単位の特殊な回転命令も追加されています。(RLD, RRD命令)。

6.各レジスタ,指定されたメモリの個々のビットを処理,チェックができること

8080で，あるレジスタ（またはメモリ）のあるビットを1にしたり，ゼロにしたりする場合は，論理演算を用いなければなりませんでした。たとえば，レジスタBの第3ビット(下位から四つ目)を1にしたければ，

A←B
A←A＞（00001000）
B←A

という三つの命令，計4バイトが必要でした。

Z80で同じことを実行すると，

SET 3, B

という2ワード命令（2バイト）で事足ります。

ビットをゼロにする場合，ビットが，1か0かチェックする場合も同様で，ダイレクトに実行できるようになっています。

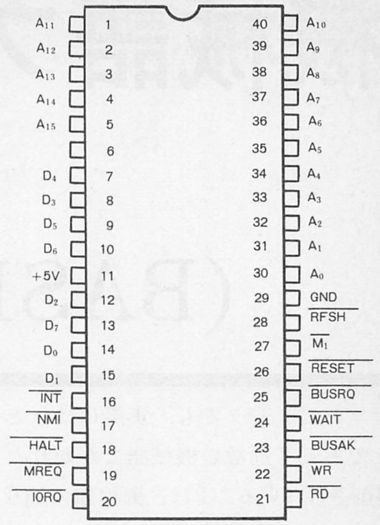
7.ブロック転送の命令のあること

大量のデータをまとめて，メモリからメモリへ移動させたり，入力ポートから受け付けたり，出力ポートへ送り出したりする命令が，Z80には用意されています。

LDIR LDDR	メモリからメモリへ
CPIR CPDR	大量データのチェック（SEARCH）
INIR INDR	入力ポートからメモリへ
OTIR OTDR	メモリから出力ポートへ

これらの命令語の下二つのアルファベット，IRまたはDRのI，D，Rの意味ですが，

IはINCREMENTのI
DはDECREMENTのD
RはREPEATのRです。



Z80外観(上面より)

これらの命令から繰り返し命令を取り除き，多少ゆうずうがきくようにした命令もあります。上との対応順に並べると

LDI, LDD, CPI, CPD, INI, IND, OUTI, OUTD

となります。

これらの命令の機能については後の章を参照してください。

8.その他

割り込み強化にともなって，必要となった，

LD A,I
LD I,A
IM 0
IM 1
IM 2

や，メモリ・リフレッシュ・レジスタRのための

LD A,R
LD R,A

レジスタAの正負を返転させる

NEG

命令などが追加されています。

機械語プログラミングのメリット

(BASICとの対比)

どんなコンピュータでも，実際には0と1の組み合わせである二進数の機械語でかかれたプログラムで制御されていることは，御承知の通りです。

しかし，0と1の組み合わせや，それを一対一の対応で記号化したアセンブラ言語と，私達が日常使っている言語やそれにともなう思考のパターンとの間には大きな隔りがあります。したがって，プログラム開発やメインテナンスの効率はいちじるしく悪くなります。

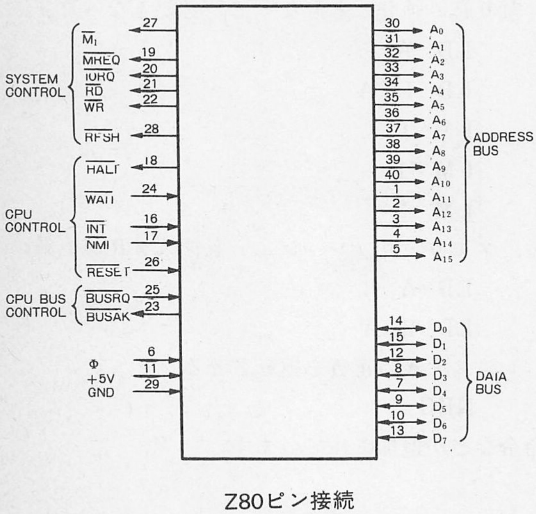
そこで一般に，人間が作るプログラムは，人間の日常語や，思考パターンに近いプログラム言語で作り，実行前，あるいは実行しながら，それらを機械語に翻訳（もちろん，コンピュータ自身に）させるという方法がとられます。

こういったプログラム言語の中に，FORTRAN, COBOL, PL／I, BASIC, PASCAL などがあるわけです。大型コンピュータでは，これらの言語を，たいていコンパイラとして，つまり，プログラムの実行前に，全プログラムを見わたして，各命令間の関係を十分に把握してから，まとめて翻訳して，それから実行する，という形で用いています。

これには，大きな容量の記憶装置が必要なため，小さなマイクロコンピュータや，パーソナルコンピュータには，あまり向いているとはいえません。

そこで，これらの小容量のコンピュータでBASICを用いる場合には，インプリンタ，つまり，命令文一行を翻訳しては実行，一行を翻訳しては実行，の形を繰り返す方法がとられるわけです。

この方法を用いますと，プログラムがコンピュータと人間の対話の中で作れるという利点もあって，プログラミングを容易で，親しみの持てるものにしてくれます。たとえば，コンパイラ型なら，全プログラムを作り終えて，それを翻訳にかけてみるまでは，文法上等のミスは発見できません。それにひきかえ対話型のものなら，入力中，実行中ともに手早く文法上のミスが発見できます。

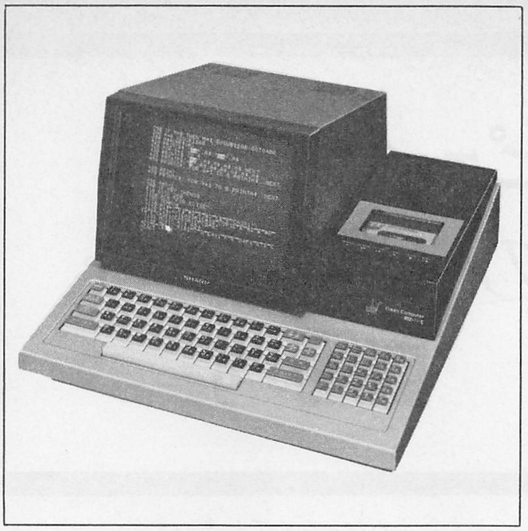


Z80ピン接続

BASICとの組合せ

こういった理由もあって，マイクロコンピュータにおけるBASICは，大ヒットし，各分野に普及したわけです。

しかし，どんなものでも慣れてくると，その限界は見えてくるもので，マイクロコンピュータのBASIC（インプリンタ）の遅さ（機械語及びコンパイラ型との比較において）と，メモリ利用効率の悪さ（特に機械語との比較において）は，いかんともしがたいもののように，思えてきます。



たとえば，マイクロコンピュータのBASICを高解像のディスプレイ制御のために使おうとするときに，その限界をまざまざと見せつけられます。

ディスプレイの最小単位の一つの点を，一つ一つ制御することによって，画面全体に変化をもたらすわけですから，高解像になればなるほど扱かわなければならないデータの数は膨大なものになります。そして，画面の左上の一点から画面の右下の一点までを制御するのに要する時間は人間の知覚できる速さより，速い方が好ましいに決まっているのですから。

また，マイクロコンピュータのBASICで，シミュレーションを行なおうとするとき，実行時内の遅いのはがまんできるものとしても，記憶容量の不足はどうしようもありません。

たとえば，1秒を1単位として，1日24時間分をシミュレートしようとすれば，60×60×24＝86400 もの記憶容量が必要となります。これをBASICの整数変数に対応できたとしても，一変数あたり5バイト（値2バイト，変数名3バイト）必要ですから，86400×5 バイトのメモリが必要になってきます。

ところが，そのシミュレーションが，1秒1単位内にある事柄が起きるか起きないかだけをチェックすればよいものであるならば，一つのメモリの一つのビットの1・0に対応できるわけです。

ということは，1バイト内に8個の記憶ができるというわけです。機械語を用いれば，メモリの節約ができるよい例です。

ちなみに，BASICの一変数が，どれくらいのバイトを必要とするか，TRS-80の場合について下にまとめておきましょう。

＊整数変数

5バイト（値2バイト，変数名3バイト）

＊単精度実数

7バイト（値4バイト，変数名3バイト）

＊倍精度実数

11バイト（値8バイト，変数名3バイト）

＊配列

最低で12バイト（型式指定により異なる）

今，述べてきた二つの例のように，実行時間を大幅に短縮したい場合や，メモリを節約したい場合には，機械語のプログラムを用いるのが有効であることは，すでにおわかりのことと思います。

しかし，先程もお話ししましたが，プログラム開発の効率や，メインテナンスの効率に関して，機械語のプログラムのそれはかなり悪いので，一般に，このような場合はBASICと機械語の混合プログラムを用いるのがよいでしょう。

つまり，時間短縮，メモリ節約に最も関りをもつ部分のみを機械語にし，その前後の処理はBASICにやらせるわけです。

マイコンのBASICに機械語のサブルーチンへジャンプし，復起する命令があるのも，そのためです。



パーソナルコンピュータで機械語のプログラムを動かす

パーソナルコンピュータを出しているメーカーの多くは，その機種に合った機械語開発用のソフトウエアパッケージを出しています。

そのソフトウエアパッケージを買い、その説明書と機械語（アセンブラ言語）の説明書を読めば，確かに機械語プログラムの開発はできます。

しかし，開発したプログラムにもよりますが，その機械語のプログラムをそのパーソナルコンピュータの中で走らせることが，容易か，といいますと，実はそうでない場合が多いのです。

開発されたプログラムが，そのパーソナルコンピュータの全システムに，完全にマッチしている場合は，容易に走らせることができるのです。しかし，そんな完璧なプログラムが一度でできるわけがありません。また，たった一個所でも，パーソナルコンピュータのシステムにマッチしない部分があれば，プログラムは暴走し，どこがどうなっていたためにそうなってしまったのか，ということすら，わからなくてしまいます。



システムにマッチしたなどという大それたプログラムでなくても，比較的小さなプログラムならＢＡＳＩＣのプログラムの中で，ＰＯＫＥ命令(直接指定のメモリへデータの書き込み)，ＰＥＥＫ命令（同データの読み取り)，ＵＳＲ命令（機械語サブルーチンの実行）などを用いて実行できるのですが，これにも，問題点が二つあります。

一つは，機械語開発用のソフトウエアパッケージを用いて作った機械語のプログラムを，直接実行することはできないということです。つまり，そのソフトウエアパッケージでは，アセンブラ言語で書かれたソースプログラムを，機械語になおし，リストを作るところまでを行ない，後は，そのリストを見て，ＢＡＳＩＣのプログラムの中へ書き込んでやらなくてはいけないわけです。

もう一つは，ＢＡＳＩＣ中で扱われる数値は，10進数だということです。機械語のプログラムは16進数で表示されますから，先に述べたリストから得られた機械語を，10進数になおしてやらなければならないわけです。だからといって，パーソナルコンピュータの他に，もう１台，トレーニング用のワンボードマイクロコンピュータを買う，というのもおかしな話しです。

初めに述べた問題点を克服するためには，そのパーソナルコンピュータのハードウエアについて十分な知識を持ち，そして，機械語プログラムに精通することしかありません。

しかし，パーソナルコンピュータからマイコンの道に入って来て，これから機械語を勉強しようとしている人に，これをいきなり要求するのも無

理な話しです。

そこで，第２点の問題点の解消に努めることになるわけです。目標を，トレーニング用のワンボードマイコンコンピュータの気軽さで，機械語プログラムの実行と評価ができること（もちろんパーソナルコンピュータにですが）とします。

機械語プログラムの実行と評価

リスト１のプログラムは，ＴＲＳ－80でこの目標達成のために作ったプログラムです。もちろん，ＢＡＳＩＣで書かれています。

このプログラムを実行すると，画面に

　　＞MEMORY　SIZE ?

と表示してきます。あらかじめ作っておいた機械語のプログラムのバイト数を数えて，その数を入力しますとすると

　　No. 1　?

と表示してきます。ここで機械語を２桁の16進数で入力します。１の場合も　０１　と入力します。

同様にして，必要な数の機械語を入力していきます。全部入力し終わると，確認のために全機械語と，それの入れられたメモリのアドレスが16進数で，ラインプリンタから打ち出されてきます。そして，画面に

　　＊＊VERIFY　THE　DATA ?
　　　ARE THEY ALL RIGHT ?
　　　1. YES　2. NO ?

と表示してきます。

入力結果にあやまりがなければ１を，あれば２をキーインします。

２を入れた場合は，画面に

　　＊NO ?

と出てきます。ミスのあった入力のナンバーを入れると（今これを３とする），

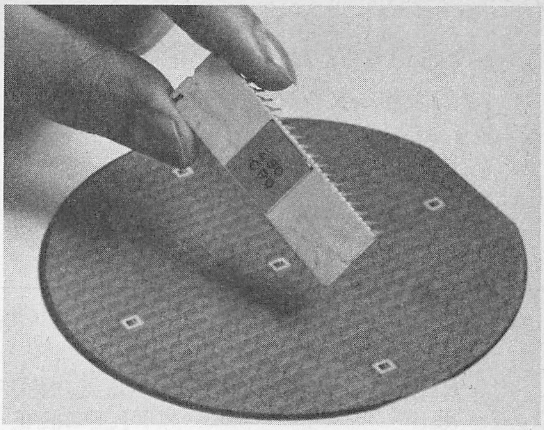
　　NO　3　＝　?

と表示してきます。ここで正しい入力をします。すると，先程と同じように

　　＊＊VERIFY　THE　DATA ?
　　　　ARE THEY ALL RIGHT ?
　　　　1. YES，　2. NO ?

と表示してきます。

１を入れると，今入力した機械語のプログラムが実行され，ひき続いて，初めに指定した分のメ



モリの内容が（ＭＥＭＯＲＹ　ＳＩＥＺ＝？のときに５を入れたのであれば，７Ｄ００から７Ｄ０４までのメモリの内容が），そのアドレスとともに16進数でラインプリンタに打ち出されてきます。

これによって，実行前のメモリの内容と，実行後のメモリの内容が比較できるわけです。

別の章で，Ｚ80の機械語の説明がしてありますが，そこに，命令の実行例があげてあります。それらはみな，このプログラムによって実行，確認，印刷されたものです。

Ｚ80以外の機種では，多少プログラムが変わるかもしれませんが，次に述べる，このプログラムの解説を十分に理解していただければ，その変更も容易にできると思われます。

プログラムの説明

それでは，**リスト１**のプログラムの説明をしていきましょう。

31　CLEAR MEM/ 10

　は未使用のメモリバイト数の1/10をストリング用に確保せよ，という命令です。16進数の入力は，まずは文字として受けつけるため，文字を入れておくメモリを確保するわけです。

32　CLS

　は画面のクリアです。

50　INPUT　“MEMORY SIZE”;PS

　は，機械語が，何バイトなのかを，ＰＳに入ておくための命令です。

80～260の命令

　では，文字入力を受けつけ，それを10000から始まるサブルーチンで数値になおして，16進アドレス7Ｄ00(注)以降のメモリに書き込む作業

（注）10進数の32000は16進数で7Ｄ00

を行ないます。ただし、×という文字と，××という文字の入力があった場合には，特別な動き方をします。×の場合は，一つ前のデータから入力のやり直しを行ないます。××の場合は，二つ前のデータから入力のやり直しを行ないます。ともに，入力した直後に気がついたミスに対して有効です。

300〜420の命令

では，16進アドレス7D00以降のメモリに入っている値を取り出し，それを16進数になおし，16進のアドレスとともに表示するという作業を行ないます。30000から始まるサブルーチン

〈リスト 1(a)〉

```
10 REM          Z80 MACHINE LANGUAGE PROGRAM TEST
20 REM                    FILE NAME -------------- Z80T/BAS
30 '
31     CLEAR MEM/10
32     CLS
40     PRINT" ** Z80 MACHINE LANGUAGE PROGRAM TEST"
50     INPUT"  > MEMORY SIZE";PS
60 '
70     PRINT"---------------------------"
80     FOR T=0 TO PS-1
90             PRINT TAB(10) "NO";T+1;"=";
100            INPUT X$
101                    IF X$="X" THEN IF T>0 THEN T=T-1 : GOTO 90
102                    IF X$="XX" THEN IF T>1 THEN T=T-2 : GOTO 90
240            GOSUB 10000'    X(DEC.) = X$(HEX.:2)
242                    IF X>255 THEN PRINT TAB(25) "????" : GOTO 90
250            POKE 32000+T,X
260    NEXT
265    PRINT"---------------------------"
270 '
300 REM                   READ
310 '
320    LPRINT" ** BEFORE RUN"
330    LPRINT"-------------------------------------"
340    FOR T=32000 TO 32000+PS-1
341            LPRINT" NO";T+1-32000,
350            X=T : GOSUB 30000
360 '                  X$(HEX.:4) = X(DEC.)
370            LPRINT X$,
380            X=PEEK(T) : GOSUB 20000
390 '                  X$(HEX.:2) = X(DEC.)
400            LPRINT X$
410    NEXT
420    LPRINT"-------------------------------------"
440 '
500 REM                   VERIFY
510 '
520    PRINT"  ** VERIFY THE DATA."
525    INPUT"          ARE THEY ALL RIGHT ?  1.YES  2.NO ",Q
530            IF Q=1 THEN 800
540            IF Q<>2 THEN PRINT"????": GOTO 525
550    PRINT"---------------------"
```

は，10進数を16進数4桁になおすサブルーチンです。20000からの方は10進数を16進数2桁になおすサブルーチンです。

500〜620の命令

では，その前の命令で表示されてきたデータにミスがないかの確認を求めてきます。そしてミスがあるとした場合には，データNO.の入力を求めてきます。それを入力すると，今度は，真のデータの入力を求めてきます。入力がすむと，300へジャンプし，また全体の確認のための表示が行なわれます。

800〜860の命令

〈リスト 1(b)〉

```
560    INPUT" * NO";T
570           IF T<1 OR T> PS THEN PRINT"????" : GOTO 560
580    PRINT TAB(10) "NO";T;"=";
590    INPUT X$
595    PRINT"-------------------------"
600    GOSUB 10000'    X(DEC.) = X$(HEX.:2)
610    POKE 32000+T-1,X
620    GOTO 300
630 '
800 REM                RUN
810 '
820    PRINT" --------  RUN  ---------"
830    DEFUSR1=&H7D00'        ( POKE 16526,0 : POKE 16527,125 )
840    X=USR1(0)'             ( X=USR(0) )
850    LPRINT" ** AFFTER RUN"
860    GOTO 330
10000 REM      SUB     X(DEC.) = X$(HEX.:2)
10010 '
10020          X1=ASC(LEFT$(X$,1)) : X2=ASC(RIGHT$(X$,1))
10030          IF X1>64 THEN X1=X1-55 ELSE X1=X1-48
10040          IF X2>64 THEN X2=X2-55 ELSE X2=X2-48
10050          X=16*X1+X2
10060  RETURN
10070 '
20000 REM      SUB     X$(HEX.:2) = X(DEC.)
20010 '
20020          X1=INT(X/16) : X2=X-16*X1
20030          IF X1>9 THEN X1=X1+55 ELSE X1=X1+48
20040          IF X2>9 THEN X2=X2+55 ELSE X2=X2+48
20050          X$=CHR$(X1)+CHR$(X2)
20060  RETURN
20070 '
30000 REM      SUB     X$(HEX.:4) = X(DEC.)
30010 '
30020          Y=INT(X/16/16) : Z=X-16*16*Y
30030          X=Y : GOSUB 20000
30040 '                 X$(HEX.:2) = X(DEC.)
30050          Y$=X$
30060          X=Z : GOSUB 20000
30070 '                X$(HEX.:2) = X(DEC.)
30080          X$=Y$+X$
30090  RETURN
30100 '
```

ここで，メモリに書き込んだ機械語が実行されます。実行後は330へジャンプし，7D00以降のメモリがもう一度表示されます。メモリ内に変化があれば，確認ができるわけです。

830　DEFUSR1＝&H7D00

USR1と名づけられた機械語のサブルーチンの先頭アドレスは，16進数の7D00である，という命令です。

ただし，これは，TRS-80のDISK BASIC用の命令で,LEVEL IIにはありません。LEVEL IIの場合は,右のカッコの中の命令を使います。

840　X＝USR1(0)

は，すでに指定されている機械語のサブルーチンを実行するときに用いるのですが，X＝の部分は，ダミーで，何の役にも立っていません。ともかく，このライン上に，USR1(0)という命令が顔を出していれば，そのサブルーチンが実行されるのです。

実行後は330へジャンプさせて，メモリの変化を確認できるようにしてあります。

10000から始まるサブルーチン

先にもお話ししましたが，文字入力を10進数にかえるサブルーチンです。

10020　X1＝ASC(LEFT$(X$，1))……

の命令は，文字変数X$の左から1番目の文字のアスキーコードをX1に入れよ，という命令です。

10030，10040　の命令

で，アスキーコードを10進数になおし，

10050　　X＝16＊X1＋X2

で，二つの文字から得られた数から，全体の値を算出するのです。

20000から始まるサブルーチン

ここでは，Xで与えられた10進数を16進数になおし，それに対応する文字のアスキーコードを計算し，最後に，文字変数として送り出します。

20020　X1＝INT(X／16)：X2＝X－16＊X1

で，10進数Xは，2桁の16進数に分解されます。X1が上の桁，X2が下の桁をうけもつことになります。

20030，20040の命令

では，数値から，それに対応する文字のアスキーコードが算出されます。

20050　X$＝CHR(X1)＋CHR(X2)

は，X1，X2に入っているアスキーコードを対応する文字になおして，二つの文字をならべたものをX$とする，という命令です。

30000から始まるサブルーチン

では，Xに入っている10進数を4桁の16進数になおす作業を行ないます。

30020　Y＝INT(X／16／16)

　　　　　　：Z＝X－16＊16＊Y

の命令でXに入っている10進数を，16進数になおした場合の上位2桁と，下位2桁を切りはなすわけです。もちろん上位の方がYで，下位の方がZです。

30030　X＝Y：GOSUB 20000

では，上位の方を，20000から始まるサブルチンで，文字になおしておきます。

30050　Y$＝X$

20000から始まるサブルーチンの入出力は，X$ですが，これは次にも使わなくてはなりません。したがって今X$の中に入っている内容をY$へ移しておくわけです。

30060　　X＝Z：GOSUB20000

今度は，下位2桁分の交換です。

30080　X$＝Y$＋X$

では，上位，下位とわけて計算および文字への変換をしたわけですが，これを二つつなげて，16進数4桁の表示になるようにするのです。

ところで，プログラム中LPRINTとあるのはラインプリンタへの出力ですが，ラインプリンタがない場合はPRINTとすれば，テレビ画面に出力されることになります。

はじめに ●●●・・・

これから　Ｚ80の各種命令の説明をしていくのですが，命令はなるべく記号で表わせるようにしておいた方が後々便利です。そこで，その記号に関する約束を初めに決めておかなければなりません。

ここでは，全体に共通するものについて述べることにしましょう。

＊ＡとＡの区別

Ｚ80にはレジスタが18個(注)あって，次のように名づけられています。

- Ｆ（フラグレジスタ）
- Ａ（アキュームレータ）
- Ｂ（レジスタＢ）
- Ｃ（レジスタＣ）
- Ｄ（レジスタＤ）
- Ｅ（レジスタＥ）
- Ｈ（Ｈレジスタ）
- Ｌ（Ｌレジスタ）
- ＩＸ（インデックスレジスタ）
- ＩＹ（インデックスレジスタ）

上のＦからＬまでには，それぞれ一つずつの予備があるため18個になります。

ところで，アセンブラプログラミングでは，16進数をよく使いますが，そこでもＡ，Ｂ，Ｃ，Ｄ，Ｅ，Ｆというアルファベットを使います。ですから，もし，プログラム上でただＡとあった場合，それが16進数のＡなのか，アキュームレータのＡなのか，わからなくなることがあります。

そこで，ここでは，次のような約束をすることにしましょう。

16進数の数値をあらわすアルファベットの前には，0をつけておく。

つまり16進数Ａ～Ｆまでは，0Ａ，0Ｂ，0Ｃ，0Ｄ，0Ｅ，0Ｆ，と表わすことにするのです。

したがって，次の記号

A←A－0A

は，アキュームレータの内容から，16進数Ａを引け，という命令になります。

また，8080，Ｚ80Ａでは，アキュームレータの内容をクリア，つまりゼロにしたいときに，

A←A－A

という命令をよく使いますが，これは，アキュームレータの内容から，自分自身の値を引け，という命令であって，16進Ａの値を引くという命令ではないので，注意してください。

＊HL≠H×L，HL＝2^8H＋L

8080やＺ80は8ビットのマイコンですが，そのアドレスは16ビットで表わすことになっています。したがって，16ビットの数を扱ったりする機会は多いので，そのための記号が必要になってきます。

(注)　インタラプト・ページ・アドレス・レジスタ（Ｉ），メモリ・リフレッシュ・レジスタ（Ｒ），スタックポインタ（ＳＰ），プログラム・カウンタ（ＰＣ）は含めていない。これらについては後で述べることにする。

普通Hとあれば，レジスタHの内容を指し，Lとあれば，レジスタLの内容を指しますが，もしHLとあったら，これは，レジスタHの内容を上位8ビット、レジスタLの内容を下位8ビットとする16ビットの数ということにします。

厳密な表し方をすれば，

2^8H+L

ということになるのですが，これをただ，

HL

と表わすことにするのです。

このようなペアには，次のようなものがあります。

BC，DE，HL，(AF)

＊A（5）←1

Z80では，8080系では不可能だった各レジスタの各ビットの制御や，チェックができます。そこで，それを表わす記号を考えなくてはいけません。

次のようにしてみました。

A(0)……レジスタAの第ゼロビット

B(7)……レジスタBの第7ビット

なお，Z80も 8080系も，第ゼロビットというのは，最下位ビットのことで，第7ビットというのは，最上位ビットのことです。

したがって，

A←0

A(5)←1

という命令をすれば，レジスタAに，十進法で表わすと 32 という数を入れることに等しいことになります。

＊CY, N, P/V, H, Z, S

Z80には6個のフラグがあります。

レジスタ全体のオーバーフローの有無をチェックする，キャリーフラグ。

計算が足し算であったか，引き算であったかチェックする　アド／サブトラクト・フラグ。

レジスタにある8個のビットの内，1になっているものの数が偶数かどうかチェックするパリティ・フラグと，レジスタ内の正の数，負の数別オーバーフローをチェックする，オーバーフロー・フラグが一緒になったもの。

8ビットを二つの4ビットに分けたとき，下位の4ビットに対して，キャリーフラグの役割をするハーフキャリーフラグ。

演算等の結果がゼロであったかどうかをチェックする。ゼロフラグ。

レジスタ内の数が負であるかどうかチェックするサインフラグ。

ですが，それぞれ

CY, N, P/V, H, Z, S

という記号で表わすことにします。

キャリーフラグは一般にCと表わすことが多いのですが，ここでは，レジスタCと区別する意味で，CYとすることにします。

また，パリティ／オーバーフロー・フラグは，P/Oと表わすこともあるのですが，

P/Oは　parity-odd（パリティが奇数である）とまちがいやすいので　P/V　とします。

また，NフラグとHフラグはソフトウェア上ででてくることはあまりありません。これは，2進数を10進数表示用に変換するDAA命令のときにCPUが必要とするものだからです。

＊A←M（IX＋X）

HLの中に入っている16進数の値をアドレスとするメモリーの内容を……　とか，インデックスレジスタIXの内容にある数を加えたときにできる値をアドレスとするメモリーの内容を……，ということがよくあります。

このように言葉で書くと，こんがらかってくることがあります。そこで，アドレス100のメモリーの内容を

M(100)

と表わすことにすると，上の二つの文章は，

M(HL)

M(IX＋X)

とを使って，シンプルな形で表現することができます。

この表現は，BASICにも通用するので色々便利だと思います。

1 ロード命令

（Z−80の全ロード命令）

Z80では，メモリーからレジスタに，数を送ったり，レジスタに直接数値を入れたり，また，レジスタからメモリーに数を送ったりする命令をすべて**ロード命令**としています。したがって，ムーブ命令や，ストア命令はありません。

ロード命令にも，扱う数値が8ビットの場合，16ビットの場合，番地修飾のし方で何通りかありますが，そういったブロックに分けて説明していくことにしましょう。

レジスタ間どうしのロード命令

レジスタBにレジスタCの内容を入れる場合などが，この命令に当ります。この場合，記号は

B←C

と書きます。アセンブラ言語では

LD B，C

と書きます。

この命令の機械語は16進数で

41

となります。

《第1−1表》 レジスタ間どうしのロード命令

LD R_1, R_2 (R_1←R_2)

		R_2						
		B	C	D	E	H	L	A
R_1	B	40	41	42	43	44	45	47
	C	48	49	4A	4B	4C	4D	4F
	D	50	51	52	53	54	55	57
	E	58	59	5A	5B	5C	5D	5F
	H	60	61	62	63	64	65	67
	L	68	69	6A	6B	6C	6D	6F
	A	78	79	7A	7B	7C	7D	7F

例．LD H, C (H←C) の機械語は 61
LD A, L (A←L) の機械語は 7D

この命令を実行すると，レジスタBにレジスタCの内容と同じものが入ります。もちろんレジスタCの内容は変化しません。また，各種フラグも変化しません。

同類の命令は全部で49個あります。

各命令と機械語の対応表を**第1-1表**にまとめておきました。

レジスタに直接 数値を入れるロード命令

レジスタCに，16進数DEを入れる場合などがこの命令に当ります。この場合，記号は

C←0DE

となります。アセンブラ言語では

LD C, 0DEH

と書きます。最後のHは前に書かれている数値が16進数(hexadecimal)であることを意味します。

この命令の機械語は

0E・DE

という2ワードにまたがるものになります。2ワード目のDEは，レジスタCに入れたい内容そのものです。

各種フラグは変化しません。

同類の命令は，7個あります。レジスタに入れる値を16進数DEとした場合，アセンブラ言語と機械語との対応について，下にまとめておきました。

LD B, 0DEH………06・DE
LD C, 0DEH………0E・DE
LD D, 0DEH………16・DE
LD E, 0DEH………1E・DE
LD H, 0DEH………26・DE
LD L, 0DEH………2E・DE
LD A, 0DEH………3E・DE

あるメモリーの内容をレジスタに入れる場合 その1 〔HLレジスタの活用〕

メモリーの番地を指定する方法が何通りかあります。ここでは，レジスタH，レジスタLをあわせてできる16ビットの数が番地を指定する場合について説明しましょう。

一例を記号で表わしてみましょう。

D←M (HL)

これは，HLの内容をアドレスとするメモリーの内容をレジスタDに入れる，という命令です。もしHに01，Lに23とあれば，HLは0123となるので，M(HL)はM(0123)ということになります。したがって上の命令は，この場合は

D←M(0123)

と同じことになります。

この命令をアセンブラ言語で表わすと

LD D, (HL)

となります。もちろんレジスタH，Lの値もメモリーM(HL)の内容も，また各種フラグの内容も変わりません。

この命令の機械語は，

56

です。

同類の命令は全部で7個あり，アセンブラ言語と機械語との対応は次のようになっています。

LD B, (HL)………46
LD C, (HL)………4E
LD D, (HL)………56
LD E, (HL)………5E
LD H, (HL)………66
LD L, (HL)………6E
LD A, (HL)………7E

あるメモリーの内容をレジスタに入れる場合 その2 〔レジスタIXの活用〕

ここでは，インデックスレジスタIXを使う場合について，説明していきます。

一例を記号で表わしてみましょう。

E←M (IX+0D)

これは，インデックスレジスタの内容に，16進数のDを加えたときに できる値を アドレスとするようなメモリーに入っている内容を，レジスタEに入れる，というものです。もし，IXに7D00という16進数が入っていたとすれば，IX+0Dは7D0Dとなるので，上の命令は，

E←M (7D0D)

と同じことになります。

この命令をアセンブラ言語で表わすと，

LD E, (IX+0DH)

となります。

もちろん，IXの値も，M(IX+0D)の内容も変わりませんし，フラグにも変化はありません。

上の命令の機械語は

DD・5E・0D

で，3ワードにまたがります。最後の0Dは，インデックスレジスタに加える値そのままです。

同類の命令は7個あって，そのアセンブラ言語との対応は次のようになっています。ただし，インデックスレジスタに加える数は，16進数のDとしておきます。

```
LD  B, (IX+0DH)
            ……DD・46・0D

LD  C, (IX+0DH)
            ……DD・4E・0D

LD  D, (IX+0DH)
            ……DD・56・0D

LD  E, (IX+0DH)
            ……DD・5E・0D

LD  H, (IX+0DH)
            ……DD・66・0D

LD  L, (IX+0DH)
            ……DD・6E・0D

LD  A, (IX+0DH)
            ……DD・7E・0D
```

ところで，あるメモリーの内容を一つのレジスタに送るのに，なぜこんな複雑な命令が必要なのか，といいますと，次のような場合に便利だからです。

〈例題〉

7D00から7D09までのメモリーに入っている内容と，7D10から7D19までのメモリーに入っている内容を一つずつとりだしてその和をもとめ，それを7D20から7D29のメモリーに入れる。

〈例解〉

この問題を比較的効率よく実行するアルゴリズムを記号で表わすと，

①IX←7D00
②B←0A
③A←M(IX+0)
④A←A+M(IX+10)
⑤M(IX+20)←A
⑥IX←IX+1
⑦B←B−1：IF B≠0 THEN GO TO③
⑧STOP

まだ説明していない命令が多く出てきますが，Z80では1行の命令は一つの命令で実行できます。⑦の命令は，レジスタBの内容から1を引いて，その結果が，ゼロでなければ，③の命令へもどれ，そうでなければ⑧の命令へ行け，というものです。個々の命令の詳しい説明は，順を追って行ないますが，インデックスレジスタ…IXの便利な点について，ある程度はわかっていただけたと思います。

あるメモリーの内容をレジスタに入れる場合　その3 〔レジスタIYの活用〕

今，インデックスレジスタIXについてお話ししました。Z80には，もう一つインデックスレジスタがあります。それはインデックスレジスタIYと名づけられています。

このIYの働きはIXと同じなので，ここでは，アセンブラ言語と機械語との対応を示すだけにしておきましょう。

```
LD  B, (IY+0EH)
            ………FD・46・0E

LD  C, (IY+0EH)
            ………FD・4E・0E

LD  D, (IY+0EH)
            ………FD・56・0E

LD  E, (IY+0EH)
            ………FD・5E・0E

LD  H  (IY+0EH)
            ………FD・66・0E

LD  L, (IY+0EH)
            ………FD・6E・0E

LD  A, (IY+0EH)
            ………FD・7E・0E
```

レジスタの内容をあるメモリーへ送る場合　その1 〔HLレジスタの活用〕

メモリーの番地を指定する方法が三つあります。ここでは，レジスタHとレジスタLを用いる方法

についてお話しします。

一例を記号で書いてみましょう。

M(HL)←E

これは，レジスタEの内容をアドレスがHLであるメモリーにも入れるという命令です。アセンブラ言語では

LD (HL), E

となります。もちろん，レジスタH，レジスタL，レジスタEの内容は変わりません。各種フラグも変わりません。

このような命令は7個ありますが，次に，そのアセンブラ言語と機械語の対応をまとめておきました。

LD (HL), B ………70
LD (HL), C ………71
LD (HL), D ………72
LD (HL), E ………73
LD (HL), H ………74
LD (HL), L ………75
LD (HL), A ………77

レジスタの内容をあるメモリーに送る場合　その2〔レジスタIXの活用〕

インデックスレジスタIXを使う場合について，お話しします。

一例を記号で書いてみましょう。

M(IX+0F)←A

このとき，もし，IXに8200が入っていたとすれば，この命令は，

M(820F)←A

と同じ働きをすることは，もうおわかりだと思います。アセンブラ言語では

LD (IX+0FH), A

となります。

もちろん，IX，Aの内容には変化ありませんし，各種フラグにも影響を与えません。

同類の命令は7個ありますが，そのアセンブラ言語と機械語の対応について，下にまとめておきました。ただし，IXに加える16進数の値はFとしてあります。

LD (IX+0FH), B
………DD・70・0F
LD (IX+0FH)
………DD・71・0F
LD (IX+0FH), D
………DD・72・0F
LD (IX+0FH), E
………DD・73・0F
LD (IX+0FH), H
………DD・74・0F
LD (IX+0FH), L
………DD・75・0F
LD (IX+0FH), A
………DD・77・0F

レジスタの内容をあるメモリーに送る場合　その3〔レジスタIYの活用〕

インデックスレジスタIYを使う場合です。考え方や機能はIXを使う場合と同じなので，アセンブラ言語と，機械語との対応をまとめておくだけにしておきましょう。

LD (IY+10H), B
………FD・70・10
LD (IY+10H), C
………FD・71・10
LD (IY+10H), D
………FD・72・10
LD (IY+10H), E
………FD・73・10
LD (IY+10H), H
………FD・74・10
LD (IY+10H), L
………FD・75・10
LD (IY+10H), A
………FD・77・10

レジスタをAに限った場合追加できるロード命令　その1

Z80などでは各種演算はレジスタA（アキュームレータ）を用いないとできません。したがって，

レジスタAに関わりを持つロード命令が，他のレジスタの場合より多くなっています。

記号，アセンブラ言語，機械語の順にまとめると，

A←M(BC)
　LD　A，(BC)　………0A

A←M(DE)
　LD，A，(DE)　………1A

M(BC)←A
　LD　(BC)，A　………02

M(DE)←A
　LD　(DE)，A　………12

以上の命令の場合，アドレスは，二つのレジスタで指定しましたが，レジスタを用いず，直接アドレスを指定する方法もあります。アドレスを指定する16ビットの上位8ビットをX，下位8ビットをYとすると，

A←M(XY)
　LD　A，(XY)……3A・Y・X

M(XY)←A
　LD　(XY)，A……32・Y・X

具体例をあげておきましょう。

A←M(7D12)　ならアセンブラ言語は，

　LD　A，(7D12H)

機械語は

　3A・12・7D

となります。7Dと12の順に注意してください。

レジスタをAに限った場合 追加できるロード命令 その2

ここでお話しするロード命令は，ちょっと特殊なもので，普通あまり使いません。使う場合には，ハードウエアの十分な知識が必要となります。

Z80は，直接，ダイナミックRAMをつなぐことができます。そのダイナミックRAM等をコントロールするレジスタ（メモリ・リフレッシュレジスタ・R）　というのがあるのですが，そのレジスタRと，レジスタAの間のロード命令があります。記号，アセンブラ言語，機械語の順でまとめると，

A←R
　LD　A，R　………ED・5F

R←A
　LD　R，A　………ED・4F

となります。

また，Z80では「割り込み」という機能が強化されています。それにともない，それを制御するレジスタ（インタラプト・ページ・アドレス・レジスタ・I）(注)というのがあります。このレジスタIと，レジスタAの間のロード命令も用意されています。記号，アセンブラ言語，機械語の順にまとめると

A←I
　LD　A，I…………ED・57

I←A
　LD　I，A　………ED・47

となります。

なお，これらの命令の活用法については，ハードウエアの十分な知識が必要となりますので，ここでは省略させていただきます。詳しいことをお知りになりたい方は，次の文献を参考にしてください。

マイクロコンピュータZ－80ハンドブック〔I〕
　4ページその他（シャープ株式会社）3,000円

（注）インタラプト・ベクタ・レジスタと呼ぶ場合もある。

16ビットのデータを レジスタ間でやりとりする ロード命令

いずれの場合も，各種フラグには影響はあたえません。記号，アセンブラ言語，機械語の順でまとめてみましょう。

SP←HL
　LD　SP，HL　………F9

SP←IX
　LD　SP，IX　………DD・F9

SP←IY
　LD　SP，IY　………FD・F9

なお，SPはスタックポインタと呼ばれ，レジスタの内容を退避されるときなどに用います。その用い方についてはPOP命令，PUSH命令のところを参考にしてください。

16ビットのデータを直接レジスタに入れるロード命令

いずれの場合も，各種フラグには影響を与えません。レジスタに入れる内容を，16進数ABCとした場合について，記号，アセンブラ言語，機械語の順でまとめてみました。

```
BC←0ABC
    LD  BC, 0ABCH
            ………… 01・BC・0A

DE←
    LD  DE, 0ABCH
            ………… 11・BC・0A

HL←0ABC
    LD  HL, 0ABCH
            ………… 21・BC・0A

SP←0ABC
    LD  SP, 0ABCH
        ………… DD・31・BC・0A

IX←0ABC
    LD  IX, 0ABCH
        ………… FD・21・BC・0A

IY←0ABC
    LD  IY, 0ABCH
        ………… FD・21・BC・0A
```

となりあう二つのメモリーが16ビットの数を表わすものと考えたとき，その値をレジスタにロードする命令

一例を記号で表わすと次のようになります。

BC←M(8201) M(8200)

これは，

B←M(8201)………上位8ビット

C←M(8200)………下位8ビット

を同時に行なうことと同じことです。BCとつなげた場合，中に16ビットの数が入っているものと考えますが，Bの方が上位8ビット，Cの方が下位8ビットとします。しかし，メモリーの方では逆で　M(8200)の方が下位8ビット，M(8201)の方が上位ビットとなりますから注意してください。

これをアセンブラ言語で書くと，

LD　BC，(8200H)

となります。下位の8ビットの入っている，アドレスの若い方を指定する点に注意してください。

この命令は，各種フラグに影響は与えません。

同類の命令を，記号，アセンブラ言語，機械語の順にまとめておきました。ただし，メモリーのアドレスは8200と8201としてあります。

```
BC←M(8201)M(8200)
    LD  BC, (8200H)
        …………ED・4B・00・82

DE←M(8201)M(8200)
    LD  DE, (8200H)
        …………ED・5B・00・82

HL→M(8201) M(8200)
    LD  HL, (8200H)
            …………2A・00・82

SP←M(8201) M(8200)
    LD  HL, (8200H)
        …………ED・7B・00・82

IX←M(8201) M(8200)
    LD  IX, (8200H)
        …………DD・2A・00・82

IY←M(8201) M(8200)
    LD  IY, (8200H)
        …………FD・2A・00・82
```

二つのレジスタが16ビットの数を表わしているとき，この数を二つのメモリーに送るロード命令

一例を記号で表わすと次のようになります。

M(7D01) M(7D00)←DE

これは，

M(7D01)←D……(上位8ビット)

M(7D00)←E……(下位8ビット)

という二つの命令を続けて行なうことと同じなのですが，DEが16ビットの数を表わしているときは，Dの方が上位8ビットで，M(7D01)M(7D00)が16ビットの数を表わしているときはM(7D01)の方が上位8ビットになるので注意して

ください。

これをアセンブラ言語で表わすと

LD (7D00H), DE

となります。下位8ビットの入るメモリーのアドレスの方を指定する点に注意してください。

もちろん，レジスタの内容，各種フラグの内容は変わりません。

同類の命令を，記号，アセンブラ言語，機械語の順にまとめると次のようになります。ただし，メモリーは，アドレス7D00，7D01 のものを使うものとしてあります。

M(7D01) M(7D00)←BC
LD (7D00H), BC
………ED・43・00・7D

M(7D01) M(7D00)←DE
LD (7D00H), DE
…………ED・53・00・7D

M(7D01) M(7D00)←HL
LD (7D00H), HL
…………………22・00・7D

M(7D01) M(7D00)←SP
LD (7D00H), SP
…………ED・73・00・7D

M(7D01) M(7D00)←IX
LD (7D00H), IX
…………DD・22・00・7D

M(7D01) M(7D00)←IY
LD (7D00H), IY
…………FD・22・00・7D

=ロード命令=
テストプログラムの作成

以上で，Z80の全ロード命令の説明が終わりました。そこで，代表的なロード命令だけを使ってプログラムを作ってみました。意味のあるプログラムではありませんが，アドレス指定法の参考にはなると思います。**第1-1図**がそれですが，このプログラムを実行すると，

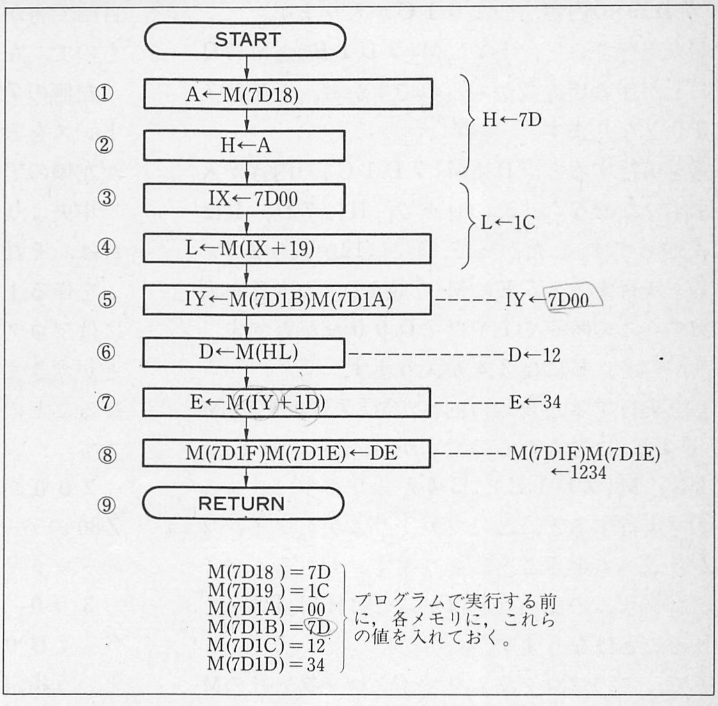
M(7D1E) に16進数32が入り，
M(7D1F) に16進数12が入る

ことになります。順を追って考えてみましょう。

①を実行すると，M(7D18) の内容，つまり7DがAに入ります。

②を実行すると，HにAの内容が入るのですから，Hは7Dとなります。

③を実行すると，インデックスレジスタIXの内容が7D00と



《第1－1図》Z80の代表的なロード命令だけを使って作ったテストプログラム

```
      MACRO-80 3.3    09-MAY-79         PAGE    1
                                  00100   ;          TEST PROGRAM (LOAD)
                                  00200   .Z80
 0000'                            00300   ASEG
                                  00400   ORG        7D00H
                                  00500   ;
 7D00     3A 7D18                 00600              LD       A,(7D18H)         ; A=M(7D18)
 7D03     67                      00700              LD       H,A               ; H=A
 7D04     DD 21 7D00              00800              LD       IX,7D00H          ; IX=7D00
 7D08     DD 6E 19                00900              LD       L,(IX+19H)        ; L=M(IX+19)
 7D0B     DD 2A 7D1A              01000              LD       IY,(7D1AH)        ; IT=M(7D1B)M(7D1A)
 7D0F     56                      01100              LD       D,(HL)            ; D=M(HL)
 7D10     FD 5E 1D                01200              LD       E,(IY+1DH)        ; E=M(IY+1D)
 7D13     ED 53 7D1E              01300              LD       (7D1EH),DE        ; M(7D1F)M(7D1E)=DE
 7D17     C9                      01400              RET                        ; RETURN
 7D18     7D                      01500              DB       7DH               ; M(7D18)=7D
 7D19     1C                      01600              DB       1CH               ; M(7D19)=1C
 7D1A     00                      01700              DB       00H               ; M(7D1A)=00
 7D1B     7D                      01800              DB       7DH               ; M(7D1B)=7D
 7D1C     12                      01900              DB       12H               ; M(7D1C)=12
 7D1D     34                      02000              DB       34H               ; M(7D1D)=34
 7D1E     00                      02100              DB       00H               ; M(7D1E)=00
 7D1F     00                      02200              DB       00H               ; M(7D1F)=00
                                  02300              END
```

《第１－２図》 アセンブラ言語でプログラムを作る

なります。

④を実行すると，インデックスレジスタＩＸの内容，ここでは７Ｄ００ですが，これに１９が加わると７Ｄ19になります。したがって，Ｌには M(７Ｄ19)の内容，つまり１Ｃが入ります。

⑤を実行すると，ＩＹにM(７Ｄ１Ｂ)とM(7D1A)とが作る16進数が入るのですから，ＩＹは７Ｄ００となります。

⑥を実行すると，ＤにM(７Ｄ１Ｃ)の内容が入ります。なぜなら，この時点で，Ｈは７Ｄ，Ｌは1Cだからです。したがって,Ｄには12が入ります。

⑦を実行すると，ＥにM(７Ｄ１Ｄ)の内容が入ります。この時点でＩＹは７Ｄ００だからです。したがって，Ｅには３４が入ります。

⑧を実行するとＤＥの内容がM(７Ｄ１Ｆ)，M(７Ｄ１Ｅ)に送られるのですから，M(７Ｄ１Ｆ)に１２，M(７Ｄ１Ｅ)に３４が入ります。

⑨を実行すると，このプログラムからメインプログラムへもどることになります。もちろんメインプログラムのないときは，ここにHALT命令を入れることになります。

さて，このプログラムをマイクロソフト社のＭ８０というソフトウエアを使って，アセンブラ言語から機械語に直してみましょう。

第１－２図が，そのときにできるリスティング・ファイルの内容です。

中央近くにある１００から23００までの数字は，右側のソースプログラム（この場合，アセンブラ言語で書かれたプログラム）を作る段階で用いるもので，左側の機械語には関係ありません。

左側の７Ｄ００から７D1Fは，機械語の入るアドレスを表わしています。その右に書かれたものが左側のアセンブラ言語に対応する機械語です。

中央より右側のリストの中で ；が書かれていれば，それより右側に書かれたものは，プログラムを作る上では無視されます。したがって，ここにはプログラムを作る者が，自由に文字を書くことができます。後からリストを見て，わからなくなることのないよう注訳を付けるときなどに便利です。

２００の右に　.Ｚ８０　とあるのは，以下，Z80のアセンブラ言語で書くという意味です（このプログラムで8080のアセンブラもできる）。

３００　４００　にかかれた，ＡＳＥＧ，ＯＲＧ　７Ｄ００　は，機械語を７Ｄ００から入れるという指定です。

５００で1行あけて，６００からがプログラムです。

マイクロソフト社のM８０に関しては，ちょっと注意しなくてはいけないところがあります。機械語のリストを見て，すでに気づかれた方もいらっしゃると思いますが，アドレス等を示す１６ビットの数値は，機械語を入力するときの順にはなっていないのです。

たとえば，７Ｄ００番地は

７Ｄ００　３Ａ　７Ｄ１８

となっていますが，実際に機械語を入力する場合には，

３Ａ・１８・７Ｄ

としなければならないのです。(注)

同じことは，７Ｄ０４，７Ｄ０Ｂ，７Ｄ13でもいえます。

７Ｄ０４　ＤＤ　２１　７Ｄ００

は，実際には

ＤＤ・２１・００・７Ｄ

といれます。

７Ｄ０Ｂ　ＤＤ　２Ａ　７Ｄ１Ａ

は

ＤＤ・２Ａ・１Ａ・７Ｄ

と入力し，

７Ｄ１３　ＥＤ　５３　７Ｄ１Ｅ

は

ＥＤ・５３・１Ｅ・７Ｄ

と入力します。

さて，６００という文番号の次に

ＬＤ　Ａ，(７Ｄ１８Ｈ)

となりますが，これはアセンブラ言語そのままです。その次に，

；　Ａ＝Ｍ(７Ｄ１８)

とあるのは　注訳です。記号

Ａ←Ｍ(７Ｄ１８)

のことなのですが ← という印字ができないためイコール（＝）を用いています。

以下　14００まで，このようにプログラムを書いていき，後でM８０というソフトウエアで処理をすると　左側のアドレスと機械語が作られてくる，というわけです。

15００以下の

ＤＢ　７ＤＨ

などの，ＤＢは，ディファイン・バイトの略で，直接，対応するアドレスへ数値を書き込むときに用いるものです。

さて，ここでプリントされてきた機械語を実際に動かしてみましょう。

Ｍ(７Ｄ１Ｅ)に　34

Ｍ(７Ｄ１Ｆ)に　12

が入るでしょうか。

プログラム実行前のメモリーの内容と，実行後のメモリーの内容を比較するものを**第１－３図**にまとめておきました。

（注）　マイクロソフト社のアセンブラ・パッケージを使っている場合，このリストを見て直接機械語をマイコンに入力することはないので，実際にはそれほど不便ではない。

** BEFORE RUN

NO 1	7D00	3A
NO 2	7D01	18
NO 3	7D02	7D
NO 4	7D03	67
NO 5	7D04	DD
NO 6	7D05	21
NO 7	7D06	00
NO 8	7D07	7D
NO 9	7D08	DD
NO 10	7D09	6E
NO 11	7D0A	19
NO 12	7D0B	FD
NO 13	7D0C	2A
NO 14	7D0D	1A
NO 15	7D0E	7D
NO 16	7D0F	56
NO 17	7D10	FD
NO 18	7D11	5E
NO 19	7D12	1D
NO 20	7D13	ED
NO 21	7D14	53
NO 22	7D15	1E
NO 23	7D16	7D
NO 24	7D17	C9
NO 25	7D18	7D
NO 26	7D19	1C
NO 27	7D1A	00
NO 28	7D1B	7D
NO 29	7D1C	12
NO 30	7D1D	32
NO 31	7D1E	00
NO 32	7D1F	00

** AFFTER RUN

NO 1	7D00	3A
NO 2	7D01	18
NO 3	7D02	7D
NO 4	7D03	67
NO 5	7D04	DD
NO 6	7D05	21
NO 7	7D06	00
NO 8	7D07	7D
NO 9	7D08	DD
NO 10	7D09	6E
NO 11	7D0A	19
NO 12	7D0B	FD
NO 13	7D0C	2A
NO 14	7D0D	1A
NO 15	7D0E	7D
NO 16	7D0F	56
NO 17	7D10	FD
NO 18	7D11	5E
NO 19	7D12	1D
NO 20	7D13	ED
NO 21	7D14	53
NO 22	7D15	1E
NO 23	7D16	7D
NO 24	7D17	C9
NO 25	7D18	7D
NO 26	7D19	1C
NO 27	7D1A	00
NO 28	7D1B	7D
NO 29	7D1C	12
NO 30	7D1D	32
NO 31	7D1E	32
NO 32	7D1F	12

《第１－３図》 第１－２図のプログラムをメモリーに書き込み
実行前と実行後の内容の変化を見たもの

2レジスタの内容を退避および交換する場合に用いる命令

これらの命令はすべて16ビット単位で行なわれます。また，各種フラグにも影響を与えないことを初めに述べておきましょう。

退避命令

マイコン内のメモリーがすべて，レジスタと同じ働きをすれば便利だな，と思うことがしばしばあります。限られたレジスタやアキュームレータを有効に使うためには，どうしても退避命令を使わなければなりません。

この退避命令は，“一時あずかり”のようなもので，ＰＵＳＨ命令と呼ばれています。

一例を記号で表わしてみましょう。

ＢＣ → Ｍ(ＳＰ－１) Ｍ(ＳＰ－２)
ＳＰ←ＳＰ－２

これは

Ｂ→Ｍ(ＳＰ－１)

Ｃ→Ｍ(ＳＰ－２)

という二つの命令を実行した後に

ＳＰ←ＳＰ－２

という命令を実行したものとみなすこともできます。

もう少し具体的にお話ししましょう。

いま，

ＳＰ＝７Ｄ０４

ＢＣ＝１２３４　　（Ｂ＝12，Ｃ＝34）

であったとしましょう。上で述べた命令を実行すると，

Ｂ→Ｍ(ＳＰ－１)　は　Ｂ→Ｍ(７Ｄ０３)

と同じことになりますから

Ｍ(７Ｄ０３)←１２

ということになります。また

Ｃ→Ｍ(ＳＰ－２)　は　Ｃ→Ｍ(７Ｄ０２)

と同じことですから，

Ｍ(７Ｄ０２)←３４

が実行されます。その後

ＳＰ←ＳＰ－２

が実行されるので

ＳＰ←７Ｄ０２

となります。

同類の命令は６個ありますが，記号，アセンブラ言語，機械語の順でまとめておきます。

ＢＣ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　BC　………C５
ＤＥ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　DE　………D５
ＨＬ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　HL　………E５
ＡＦ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　AF　………F５
ＩＸ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　IX　………DD・E５
ＩＹ→Ｍ(SP－1)Ｍ(SP－2)，SP←SP－２
　PUSH　IY　………FD・E５

退避解除命令

これは，ＰＵＳＨ命令で退避させた内容をもとにもどす場合に有効な命令です。また，8080やＺ80には16ビットの内容をレジスタからレジスタに送るロード命令は限られたレジスタ間でしかできません。こんな場合に，ＰＵＳＨで退避させておいて，退避を解除する場合に，別のレジスタにもどしてやる，という利用法もあります。

この退避解除命令はＰＯＰ命令と呼ばれています。一例を記号で書いてみましょう。

　ＢＣ←Ｍ(ＳＰ＋１) Ｍ(ＳＰ)，
　　ＳＰ←ＳＰ＋２

具体的な例でやってみましょう。先程，PUSH命令のところで

　ＳＰ＝７Ｄ０４
　ＢＣ＝１２３４

のときに

　ＰＵＳＨ　ＢＣ

を実行しました。その後で

　Ｍ(７Ｄ０２)＝34
　Ｍ(７Ｄ０３)＝12
　ＳＰ＝７Ｄ０２

となりました。ここで

　ＰＯＰ　ＢＣ

を実行してみましょう。ＳＰ＝７Ｄ０２ですからこれは

　ＢＣ←Ｍ(７Ｄ０３)Ｍ(７Ｄ０２)
　　ＳＰ←７Ｄ０２＋２

ということになります。Ｍ(7Ｄ03)には12が，Ｍ(7Ｄ０２)には34が入っていますから，

　ＢＣ＝１２３４

となって，その後

　ＰＣ←ＰＣ＋２　……

が実行されるので

　ＰＣ＝７Ｄ０４

となって，すべて，もとどうりになりました。

ここで

　ＰＯＰ　ＢＣ

のかわりに

　ＰＯＰ　ＩＸ

とすれば，はじめＢＣに入っていた内容を　ＩＸに移すこともできるわけです。

同類の命令をまとめておきましょう。

ＢＣ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　BC　………C１
ＤＥ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　DE　………D１
ＨＬ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　HL　………E１
ＡＦ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　AF　………F１
ＩＸ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　IX　………DD・E１
ＩＹ←Ｍ(SP－1)Ｍ(SP)，SP←SP＋２
　POP　IY　………FD・E１

エクスチェンジ命令 その１〔レジスタ間の場合〕

２組のレジスタに16ビットのデータが入っているときに，その二つを入れかえる命令です。

Ｚ80には，各レジスタが一つの予備を持っているのですが，それを入れかえる場合も，この命令を使います。

一例を記号で表わしてみましょう。

　DE←→HL

もし，ＤＥに１２３４，ＨＬに５６７８が入っていたとして，この命令を実行すると，ＤＥの方に５６７８が，ＨＬの方に１２３４が入るわけです。

前にも述べましたが，各種フラグは影響をうけ

ません。

この命令をアセンブラ言語で表わすと，

EX DE, HL

となります。

同類の命令を下にまとめてみました。

DE⟷HL
EX DE, HL ………EB

AF⟷A'F' (A'はAの予備レジスタ)
EX AF, AF'………08

BC⟷B'C', DE⟷D'E', HL⟷H'L'
EEX ……………………D9

なお，アセンブラ言語では，AF'と書かれていても，実際の意味は A'F' ですから注意してください。また記号からおわかりだと思いますが，EEX命令は，同時に3組の交換をするので，注意してください。

エクスチェンジ命令その2
〔レジスタとメモリーの間の場合〕

メモリーのアドレス指定は，スタックポインタを使って行ないます。

一例を記号で表わしてみましょう。

M(SP+1) M(SP)⟷HL

HLではHが上位8ビット，Lが下位8ビットなのに対し，M(SP+1)M(SP) ではM(SP+1)が上位8ビット，M(SP)が下位8ビットです。

これをアセンブラ言語で表わすと

EX (SP), HL

となります。

同類の命令を，記号，アセンブラ言語，機械語の順でまとめてみましょう。

M(SP+1) M(SP)⟷HL
EX (SP), HL ………E3

M(SP+1) M(SP)⟷IX
EX (SP), IX ………DD・E3

M(SP+1) M(SP)⟷IY
EX (SP), IY ………FD・IY

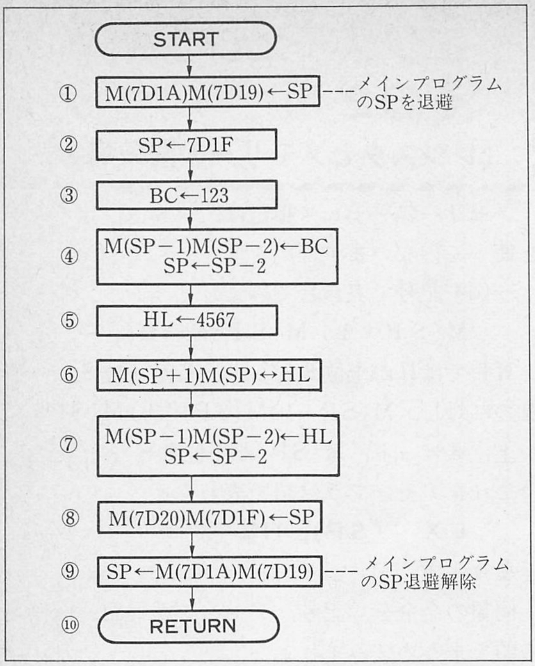
退避命令，エクスチェンジ命令をテストするプログラムの作成

Z80における退避命令や，エクスチェンジ命令を一通り説明しました。そこで，これらのうち代表的なものを使ってプログラムを考えてみることにしましょう。

第2−1図のようなものを考えてみましたが，特に意味ある仕事を効率よくする，とかいったものではありません。これを通して，スタックポインタの変化や，メモリーの変化を見ようというものです。

これを実行すると，

M(7D1B)=23
M(7D1C)=01
M(7D1D)=67
M(7D1E)=45
M(7D1F)=1B
M(7D20)=7D



《第２－１図》 PUSH命令，ＥＸ命令をテストするプログラム

となります。これがわかれば，この命令も卒業といえるでしょう。順を追って説明していきましょう。

①の命令は，スタックポインタをM(7D１９)，M(７D１A)に退避させるために設けたものです。あるメインプログラム中に，このテストプログラムを実行しようとする場合には，このようにしておかなければなりません。メインプログラムからサブルーチンへ飛んだり，もどったりする場合にスタックポインタを用いるからで，したがってサブルーチンの中でスタックポインタの値を不規則に変えると，メインプログラムに帰れなくなるのです。

②で，このサブルーチン専用のスタックポインタを設定します。

③ＢＣに０１２３を入れます。

④ＢＣの内容を　スタックポインタで退避させます。この場合は退避させて，ＢＣに別の仕事をさせるというわけではありませんが，これでＢＣはフリーになりました。ここで，

M(７D１E)に０１　が

M(７D１D)に２３　が

入ります。スタックポインタの内容は２が引かれるので，

ＰＣ＝７Ｄ１Ｄ

となっています。

⑤ＨＬに４５６７が入ります。

⑥M(７D１E) M(７D１D)⟷ＨＬが実行され

M(７D１E)に４５

M(７D１D)に６７

が入ります。そして

ＨＬには０１２３

が入ります。

⑦ＨＬの内容，つまり０１と２３がスタックポインタによってM(７D１C)とM(７D１B)に移されます。そして，スタックポインタの内容から２が引かれるのでこの段階で

M(７D１C)＝０１

M(７D１B)＝２３

ＳＰ＝７Ｄ１Ｂ

となります。

⑧ここで，スタックポインタの内容，つまり７Ｄ１ＢがM(7D２０)とM(7D1F)に移されます。したがってこの命令を実行し終わった時点で，

M(７D２０)＝７D

M(７D１F)＝１B

となります。

⑨ここで，先程退避させておいたメインプログラムにおけるスタックポインタの内容をもとにもどします。

⑩これでサブルーチンを終わりとし，メインプログラムにもどります。

以上の説明でおわかりのように，このプログラム全部を終了した時点では，

M(７D１B)＝２３

M(７D１C)＝０１

M(７D１D)＝６７

M(７D１E)＝４５

M(７D１F)＝１B

M(７D２０)＝７D

が入ります。M(7D19)，M(7D1A)には，メインプログラム中でのスタックポインタの内容が入ってくるはずです。これは，メインプログラムがどんなものかによって変わってきますので，ここでは何とはいえません。

このプログラムをアセンブラ言語で作り，それをマイクロソフト社のM80にかけて，機械語を

```
      MACRO-80 3.3    09-MAY-79        PAGE    1
                                00100   ,       TEST PROGRAM ( EX, PUSH, )
                                00200   .Z80
 0000'                          00300   ASEG
                                00400   ORG     7D00H
                                00500   ;
 7D00    ED 73 7D19             00600           LD      (7D19H),SP      ; M(7D1A)M(7D19)=SP
 7D04    31 7D1F                00700           LD      SP,7D1FH        ; SP=7D1F
 7D07    01 0123                00800           LD      BC,123H         ; BC=123
 7D0A    C5                     00900           PUSH    BC              ; M(SP-1)M(SP-2)=BC : SP=SP-2
 7D0B    21 4567                01000           LD      HL,4567H        ; HL=4567
 7D0E    E3                     01100           EX      (SP),HL         ; M(SP+1)M(SP) <-> HL
 7D0F    E5                     01200           PUSH    HL              ; M(SP-1)M(SP-2)=HL : SP=SP-2
 7D10    ED 73 7D1F             01300           LD      (7D1FH),SP      ; M(7D20)M(7D1F)=SP
 7D14    ED 7B 7D19             01400           LD      SP,(7D19H)      ; SP=M(7D1A)M(7D19)
 7D18    C9                     01500           RET                     } RETURN
                                01600           END
```

《第2－2図》 第2－1図のプログラムをアセンブラ言語，機械語になおしたもの

** BEFORE RUN			** AFFTER RUN		
NO 1	7D00	ED	NO 1	7D00	ED
NO 2	7D01	73	NO 2	7D01	73
NO 3	7D02	19	NO 3	7D02	19
NO 4	7D03	7D	NO 4	7D03	7D
NO 5	7D04	31	NO 5	7D04	31
NO 6	7D05	1F	NO 6	7D05	1F
NO 7	7D06	7D	NO 7	7D06	7D
NO 8	7D07	01	NO 8	7D07	01
NO 9	7D08	23	NO 9	7D08	23
NO 10	7D09	01	NO 10	7D09	01
NO 11	7D0A	C5	NO 11	7D0A	C5
NO 12	7D0B	21	NO 12	7D0B	21
NO 13	7D0C	67	NO 13	7D0C	67
NO 14	7D0D	45	NO 14	7D0D	45
NO 15	7D0E	E3	NO 15	7D0E	E3
NO 16	7D0F	E5	NO 16	7D0F	E5
NO 17	7D10	ED	NO 17	7D10	ED
NO 18	7D11	73	NO 18	7D11	73
NO 19	7D12	1F	NO 19	7D12	1F
NO 20	7D13	7D	NO 20	7D13	7D
NO 21	7D14	ED	NO 21	7D14	ED
NO 22	7D15	7B	NO 22	7D15	7B
NO 23	7D16	19	NO 23	7D16	19
NO 24	7D17	7D	NO 24	7D17	7D
NO 25	7D18	C9	NO 25	7D18	C9
NO 26	7D19	00	NO 26	7D19	DF
NO 27	7D1A	00	NO 27	7D1A	7B
NO 28	7D1B	00	NO 28	7D1B	23
NO 29	7D1C	00	NO 29	7D1C	01
NO 30	7D1D	00	NO 30	7D1D	67
NO 31	7D1E	00	NO 31	7D1E	45
NO 32	7D1F	00	NO 32	7D1F	1B
NO 33	7D20	00	NO 33	7D20	7D

《第2－3図》 第2－2図のプログラムの実行前、実行後のメモリーの比較

出したものを**第2－2図**に示しておきました。このリストの見方，注意すべき点については，ロード命令の終わりのところで，お話ししましたので，もう一度，確認をしておいてください。

第2－3図は，この機械語をメモリーに入れ，実際に実行する前後のメモリーの変化を見たものです。先程の通りになっているのがおわかりでしょう。

ここでは実験をしませんでしたが，

AF←→A'F' ……EX　AF, AF'

を使うとフラグの内容の退避ができ，したがって各種条件判断を必要な分だけ後にのばして実行することができます。

これは条件判断命令を説明するときに，詳しくお話しすることにしましょう。

3　8ビットの演算命令

8ビットどうしの間で足したり引いたり，あるいは論理和を求めたりする命令です。一般に，これらの命令はレジスタA（アキュームレータ）を介して行なわれます。

また，これらの命令では必ず，各種フラグに影響を与えますので，注意してください。ただし，前にもお話ししましたが，Nフラグ（アド／サブトゥラクト・フラグ）と，Hフラグ（ハーフ・キャリーフラグ）の変化をとらえて条件判断をすることは，ソフトウエア上ではできません。したがって，ここでは，それら二つのフラグの変化については記さないことにします。

レジスタAと他のレジスタとの和

一例を記号で書くと

A←A＋B

となります。これはレジスタAに入っている内容と，レジスタBに入っている内容をとりだしてきて，これらの和を求め，その結果をレジスタAに入れるという命令です。

アセンブラ言語では

ADD　A, B

と書きます。このときの機械語は

80

です。

各種フラグの変化についてまとめてみましょう。

S ： 足し算の結果，最上位のビット（第7ビット）が1になると，S＝1，そうでなければ，S＝0

Z ： 足し算の結果，全ビットがゼロになったら，Z＝1，そうでなければ，Z＝0

P/V： 正の数どうしの和がオーバーフローを起こし，負の数と同じ表示になってしまったり，負の数どうしの和が同じように，正の数と同じ表示になってしまったときに P/V=1，そうでなければ P/V＝0

CY： 足し算の結果が8ビット以上の数になった場合 CY＝1，そうでなければ CY＝0

同類の命令は7個あります。記号，アセンブラ言語，機械語の順にまとめておきました。

```
A←A＋B
  ADD  A，B    ………80

A←A＋C
  ADD  A，C    ………81

A←A＋D
  ADD  A，D    ………82

A←A＋E
  ADD  A，E    ………83

A←A＋H
  ADD  A，H    ………84

A←A＋L
  ADD  A，L    ………85

A←A＋A
  ADD  A，A    ………87
```

レジスタAに直接数値を加える命令

記号で表わすと

A←A＋X　　　（Xは8ビットの数）

となります。Xが16進数23のときのアセンブラ言語，機械語は

```
ADD  A，23H
   ………C6・23
```

となります。

この命令が，各フラグに与える影響は，レジスタAと他のレジスタの間の和を求める場合と同じですから，そこを参照してください。

レジスタAと あるメモリーの内容の和を求める命令

これを実行するのに，三つの方法があります。一つは，HLレジスタを使う方法，もう一つは，IXレジスタを使う方法，そして最後はIYレジスタを使う方法です。

記号，アセンブラ言語，機械語の順にまとめてみましょう。

```
A←A＋M（HL）
  ADD  A，（HL）…………86

A←A＋M（IX＋X）
  ADD  A，（IX＋X）
            …………DD・86・X

A←A＋M（IY＋X）
  ADD  A，（IY＋X）
            …………FD・86・X
```

ただし，Xは8ビットの数値です。Xが16進数のABなら，

```
A←A＋M（IX＋0ABH）
  ADD  A，（IX＋0ABH）
               DD・86・AB
```

というふうになります。

これらの命令が，各フラグに与える命令は，レジスタAと他のレジスタの和を求める命令の場合と同じですので，そこを参照してください。

レジスタAと他のレジスタの和にキャリーの値を加える命令

一例を記号で書くと

A←A＋B＋CY

となります。なぜこのような命令があるか，といいますと，たとえば，2組の16ビットの数の和を求めるときに便利だからです。それは次のようにすればよいでしょう。16ビットの数はレジスタBC，レジスタDEに入っているものとします。

① A←C

② A←A＋E

③ C←A

④ A←B

⑤ A←A＋D＋CY

⑥ B←A

これは

BC←BC＋DE

と同じことになります。

②で二つの下位8ビットの和を求めているのですが，その結果，もし8ビットをオーバーするようなことがあれば，それは，キャリーフラグCYに保存されます。③④の命令はCYには影響を与えないことをすでに学びました。したがって，オーバーした1ビットがあれば，⑤で，上位8ビットの和を求めるときに一緒に加えておけばよいのです。

このような命令は全部で，7個あります。記号，アセンブラ言語，機械語の順でまとめてみましょう。

A←A＋B＋CY
ADC　A，B　…………88

A←A＋C＋CY
ADC　A，C　…………89

A←A＋D＋CY
ADC　A，D　…………8A

A←A＋E＋CY
ADC　A，E　…………8B

A←A＋H＋CY
ADC　A，H　…………8C

A←A＋L＋CY
ADC　A，L　…………8D

A←A＋A＋CY
ADC　A，A　…………8F

この演算のフラグに与える影響は次の通りです。

S　：　この演算の結果，最上位のビット（第7ビット）が　1ならば，S＝1，そうでなければ，S＝0

Z　：　この演算の結果，すべてのビットがゼロならば，Z＝1，そうでなければZ＝0

P/V：　正の数どうしの和が　負，負の数どうしの和が正といった表示になってしまう場合　P/V＝1，そうでなければP/V＝0

CY：　この演算の結果が8ビットを越えるとき，CY＝1，そうでなければCY＝0

レジスタAに直接数値を加え、それにキャリーの値を加える命令

8ビット以下で表わせる，16進数の数をXとすると，記号は次のようになります。

A←A＋X＋CY

これをアセンブラ言語で書くと

ADC　A，X

となります。

機械語は，

CE・X

となります。XがABなら，アセンブラ言語，機械語は

ADC　A，0ABH………CE・AB

となります。

この命令が各フラグに与える影響は，レジスタAと他のレジスタの和にCYの値を加える命令の場合と同じですので，そこを参照してください。

レジスタAと　あるメモリーの和にキャリーの内容を加える命令

この命令の必要性については，前のところでおわかりだと思います。この命令には，三つの種類があります。HLレジスタを使うもの，IXレジスタを使うもの，IYレジスタを使うものという具合です。IXやIYに加える数は16進数Fとして，まとめてみました。

A←A＋M(HL)＋CY
ADC　A，(HL)　………8E

A←A＋M(IX＋0F)
ADC　A，(IX＋0FH)
………　DD・8E・0F

A←A＋M(IY＋0F)
ADC　A，(IY＋0FH)
………　FD・8E・0F

この演算の，各種フラグへ与える影響は，レジ

スタAと他のレジスタの和にCYの値を加える場合と同じですので，そこを参照してください。

レジスタAと他のレジスタの間の差を求める命令

一例を記号で表わしてみましょう。

A←A－B

これをアセンブラ言語で表わすと，

SUB　B

となります。注意してください。

SUB　A，B

ではありませんから。特に引き算の場合は，必ずレジスタAを介して行なわれます。したがってAを省いても，まちがえることはない，ということで，そうしたのでしょう。

同類の命令は7個あります。下にまとめておきました。

A←A－B
SUB　B …………90

A←A－C
SUB　C …………91

A←A－D
SUB　D …………92

A←A－E
SUB　E …………93

A←A－H
SUB　H …………94

A←A－L
SUB　L …………95

A←A－A
SUB　A …………97

この命令の実行結果が，各フラグへ与える影響は次の通りです。

S：　この演算の結果，最上位のビット（第7ビット）が1になれば，S＝1，そうでなければ，S＝0

Z：　この演算の結果がゼロならば，Z＝1，そうでなければ，Z＝0

P/V：　負の数から正の数を引いたときに，正の数と同じ表現になってしまったり，正の数から，負の数を引いたときに，負の数と同じ表現になってしまったときに，P/V＝1，そうでなければP/V＝0

CY：　8ビットの表わしている数がすべて正であるとした場合，つまり，8ビットが10進数で表わせば0～255の間の数を表わしているものとみなした場合に引く方の数が，引かれる方の数より大きい場合，CY＝1，そうでなければ，CY＝0

たとえば，次の数値が16進数のとき

FB－FF　→　CY＝1
73－74　→　CY＝1
62－81　→　CY＝1

となります。

レジスタAから直接数値を引く場合の命令

16進数で2桁の数をXとすると，記号は次のようになります。

A←A－X

これをアセンブラ言語で表わすと，次のようになります。

SUB　X

機械語は，

D6・X

です。XがBCなら，アセンブラ言語，機械語は次のようになります。

SUB　0BCH …………D6・BC

この命令が各フラグに与える影響は，レジスタAと他のレジスタの間の差を求める場合と同じですので，そこを参照してください。

レジスタAからあるメモリーの内容を引く場合の命令

メモリーのアドレスを指定する方法によって，三つの種類があります。それぞれ，記号，アセンブラ言語，機械語の順でまとめてみましょう。

```
A←A－M（HL）
   SUB  （HL）  …………96

A←A－M（IX＋10）
   SUB  （IX＋10H）
            ………………DD・96・10

A←A－M（IY＋10）
   SUB  （IY＋10H）
            ………………FD・96・10
```

ここでは，ＩＸやＩＹに加える値を16進数の10としてあります。

この命令の実行結果が，各種フラグに与える影響は，レジスタAと他のレジスタの間の差を求める場合と同じですから，そこを参照してください。

レジスタAと他のレジスタの差からキャリーの値を引く場合の命令

一例を記号で表わしてみましょう。

A←A－B－CY

この命令も　2ワード以上にまたがる数の引き算のときに，重要な役割を果たします。

たとえば，BCレジスタ，DEレジスタに，それぞれ16ビット以内で表わせる数が入っているときに，

BC←BC－DE

を行なう場合には，次のようにします。

① A←C

② A←A－E

③ C←A

④ A←B

⑤ A←A－D－CY

⑥ B←A

②の演算で　下位8ビットの演算をしますが，Eの方が大きれば，ここでCYに1がセットされます。③④において，CYは変化しませんから，この値は⑤に送られ，そこで，AとDの差を求めるときに一緒に処理されるというわけです。

同種の命令は7個あり，まとめると次のようになります。

```
A←A－B－CY
   SBC  A，B  …………98

A←A－C－CY
   SBC  A，C  …………99

A←A－D－CY
   SBC  A，D  …………9A

A←A－E－CY
   SBC  A，E  …………9B

A←A－H－CY
   SBC  A，H  …………9C

A←A－L－CY
   SBC  A，L  …………9D

A←A－A－CY
   SBC  A，A  …………9F
```

この演算が，各フラグに与える影響は，次の通りです。

- S：　演算の結果，最上位のビット（第7ビット）が1ならば，S＝1，そうでなければ，S＝0
- Z：　演算の結果，8個のビットすべてがゼロになった場合に，Z＝1，そうでなければ，Z＝0
- P/V：　負の数から正の数を引いたときに，結果が正の数と同じ表示になってしまったり，正の数から負の数を引いたときの結果が負の数と同じ表示になってしまうときに，P/V＝1，そうでないときにP/V＝0
- CY：　A←A－B－CY　の場合，8ビットで表わされる数が0～255であるとみなしたときに，A≦（B＋CY）なら，CY＝1，そうでなければ，CY＝0

レジスタAから直接数値を引き、さらにCYの値を引く命令

16進数で2桁の数をXとすると，記号は

A←A－X－CY

となります。アセンブラ言語の方は

SBC A，X

となり，機械語は，

DE・X

です。Xが76なら，アセンブラ言語，機械語は次のようになります。

SBC 76H ……………DE・76

この命令が各フラグに与える影響は，レジスタAと他のレジスタの差から，CYの値を引く場合と同じですから，そこを参照してください。

レジスタAと あるメモリーの内容の差からキャリーの値を引く場合の命令

この命令の役割は，レジスタ間どうしの引き算の場合と同じです。

メモリーのアドレスを指定する方法のちがいで，三つの命令があります。以下まとめてみましょう。

ただし，IXレジスタ，IYレジスタに加える数を16進数の12としてあります。

A←A－M（HL）－CY
SBC A，（HL）………9E

A←A－M（IX＋12）
SBC A，（IX＋12）
…………DD・9E・12

A←A－M（IY＋12）
SBC B，（IY＋12）
…………FD・9E・12

この演算が，各フラグに与える影響は，レジスタAと他のレジスタの差から，CYの値を引く場合と同じですから，そこを参照してください。

ADC，SBC命令を使ったテストプログラムの作成

各命令を説明するところでお話しした，2ワードにまたがる数の計算をしてみることにしましょう。

BC←BC＋DE

という計算を

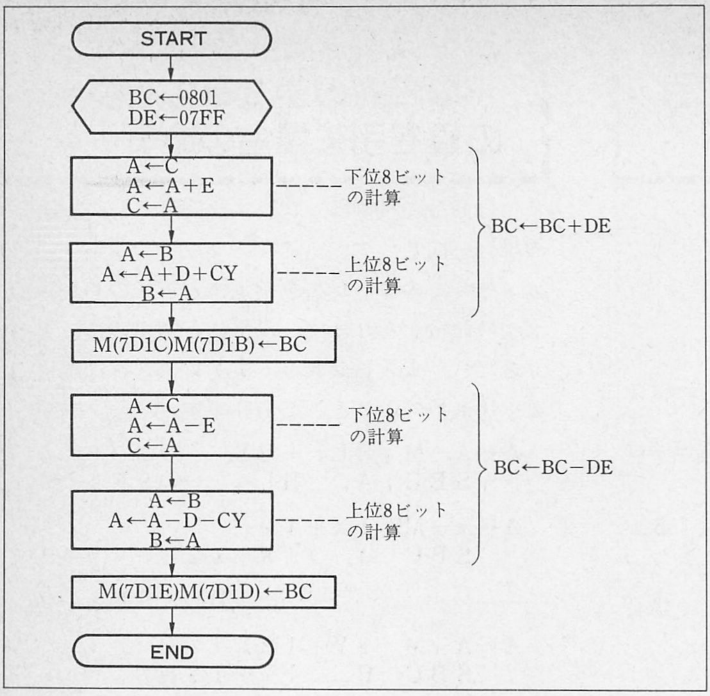
BC←BC－DE

という計算ができるプログラムを作ってみるのですが，**第3－1図**のようなプログラムを作ってみました。

はじめBCには0801，DEには07FFが入っています。ここで

BC←BC＋DE

を計算するのですが，プログラムがうまく働らけば，BCは



《第3－1図》
ADC，SBC命令のテストプログラム

```
MACRO 80 3.3    09-MAY-79         PAGE    1
                                  ;         TEST PROGRAM ( ADC, SBC )
                                  .Z80
0000'                             ASEG
                                  ORG       7D00H
                                  ;
7D00    01 0801                             LD        BC,801H          ; BC=0801H
7D03    11 07FF                             LD        DE,7FFH          ; DE=07FFH
7D06    79                                  LD        A,C              ; A=C
7D07    83                                  ADD       A,E              ; A=A+E
7D08    4F                                  LD        C,A              ; C=A
7D09    78                                  LD        A,B              ; A=B
7D0A    8A                                  ADC       A,D              ; A=A+D+CY
7D0B    47                                  LD        B,A              ; B=A
7D0C    ED 43 7D1B                          LD        (ADR1),BC        ; M(ADR1+1)M(ADR1)=BC
                                  ;
7D10    79                                  LD        A,C              ; A=C
7D11    93                                  SUB       E                ; A=A-E
7D12    4F                                  LD        C,A              ; C=A
7D13    78                                  LD        A,B              ; A=B
7D14    9A                                  SBC       A,D              ; A=A-D-CY
7D15    47                                  LD        B,A              ; B=A
7D16    ED 43 7D1D                          LD        (ADR2),BC        ; M(AD2+1)M(ADR2)=BC
7D1A    C9                                  RET                        ; RETURN
7D1B    00                        ADR1.     DB        00H              ; M(ADR1)=0
7D1C    00                                  DB        00H              ; M(ADR1+1)=0
7D1D    00                        ADR2.     DB        00H              ; M(ADR2)=0
7D1E    00                                  DB        00H              ; M(ADR2+1)=0
                                            END
```

《第3－2図》　第3－1図のフローチャートをアセンブラプログラミングしたもの

　　０８０１
＋　０７ＦＦ
　　１０００

より，１０００になるはずです。これを，メモリー７Ｄ１Ｃと７Ｄ１Ｂに送った後，今度は，

ＢＣ←ＢＣ−ＤＥ

を計算します。プログラムが正しければ，ＢＣはもとにもどるわけですから，０８０１になるはずです。これをメモリー７Ｄ１Ｅと７Ｄ１Ｄに送ります。以上のことから，結果として

Ｍ(７Ｄ１Ｂ)＝００ ⎫
Ｍ(７Ｄ１Ｃ)＝１０ ⎭ １０００
Ｍ(７Ｄ１Ｄ)＝０１ ⎫
Ｍ(７Ｄ１Ｅ)＝０８ ⎭ ０８０１

のようになることが，おわかりだと思います。

アセンブラによるプログラム例を**第３−２図**にあげておきました。注意すべき点等につきましては，ロード命令のテストプログラムのところを参照してください。

ここでは，シンボルの使い方に注目しておいてください。

７Ｄ１Ｂ　の代りに　ＡＤＲ１
７Ｄ１Ｄ　の代りに　ＡＤＲ２

を使っています。アセンブラプログラミング中にアドレスが確認できないときに便利な方法です。

このリストより，直接Ｚ80に機械語を入れ，実行したものが，**第３−３図**です。ここでは，実行前のメモリーの内容と，実行後のメモリーの内容が，確認できるようになっています。

```
** BEFORE RUN
----------------------------------------
NO 1        7D00        01
NO 2        7D01        01
NO 3        7D02        08
NO 4        7D03        11
NO 5        7D04        FF
NO 6        7D05        07
NO 7        7D06        79
NO 8        7D07        83
NO 9        7D08        4F
NO 10       7D09        78
NO 11       7D0A        8A
NO 12       7D0B        47
NO 13       7D0C        ED
NO 14       7D0D        43
NO 15       7D0E        1B
NO 16       7D0F        7D
NO 17       7D10        79
NO 18       7D11        93
NO 19       7D12        4F
NO 20       7D13        78
NO 21       7D14        9A
NO 22       7D15        47
NO 23       7D16        ED
NO 24       7D17        43
NO 25       7D18        1D
NO 26       7D19        7D
NO 27       7D1A        C9
NO 28       7D1B        00
NO 29       7D1C        00
NO 30       7D1D        00
NO 31       7D1E        00
----------------------------------------
```

```
** AFFTER RUN
----------------------------------------
NO 1        7D00        01
NO 2        7D01        01
NO 3        7D02        08
NO 4        7D03        11
NO 5        7D04        FF
NO 6        7D05        07
NO 7        7D06        79
NO 8        7D07        83
NO 9        7D08        4F
NO 10       7D09        78
NO 11       7D0A        8A
NO 12       7D0B        47
NO 13       7D0C        ED
NO 14       7D0D        43
NO 15       7D0E        1B
NO 16       7D0F        7D
NO 17       7D10        79
NO 18       7D11        93
NO 19       7D12        4F
NO 20       7D13        78
NO 21       7D14        9A
NO 22       7D15        47
NO 23       7D16        ED
NO 24       7D17        43
NO 25       7D18        1D
NO 26       7D19        7D
NO 27       7D1A        C9
NO 28       7D1B        00
NO 29       7D1C        10
NO 30       7D1D        01
NO 31       7D1E        08
----------------------------------------
```

《第３−３図》　第３−２図のプログラムをメモリーに書き込み，実行前と実行後の内容の確認をしたもの

レジスタAと他のレジスタの間のAND命令

一例を記号で表わすと

A←A∧E

となります。今，レジスタA，Eの各ビットが次のようになっていたとします。

A＝01101011……(6B)
E＝01000001……(41)

この場合，AとEの両方で1になっているビットは第ゼロビット（最下位ビット）と，第6ビットだけです。したがって　A∧E　という演算を行なうと，この第ゼロビットと第6ビット以外のビットはすべてゼロになり，

01000001……(41)

となり，これがAに入ります。こういった演算をANDと呼んでいるのです。

この演算の命令は，数の処理のときにはあまり使いませんが，入出力装置に出し入れする各種のコードを処理するときには使います。

同じような命令を，記号，アセンブラ言語，機械語の順でまとめておきました。

A←A∧B
AND　B　………………　A0

A←A∧C
AND　C　………………　A1

A←A∧D
AND　D　………………　A2

A←A∧E
AND　E　………………　A3

A←A∧H
AND　H　………………　A4

A←A∧L
AND　L　………………　A5

A←A∧A
AND　A　………………　A7

この演算が，各フラグに与える影響を次にまとめてみましょう。特に P/V フラグの変化には注意してください。

S　：　演算の結果，最上位のビット（第7ビット）が1ならば，S＝1.そうでなければ　S＝0

Z　：　演算の結果，すべてのビットがゼロになれば，Z＝1.　そうでなければ　Z＝0

P/V：　演算の結果，8個のビットの内，1になっているビットの数が偶数ならP/V＝1,そうでなければ，P/V＝0

CY：　どのような場合も　ゼロ

レジスタAと直接ある数値とのANDをとる命令

16進で表わされた数をXとすると

A←A∧X

ということになります。アセンブラ言語は

AND　X

機械語は

E6・X

となります。

Xが78なら，アセンブラ言語，機械語は次のようになります。

AND　78H　………　E6・78

この命令が各フラグに与える影響は，レジスタAと他のレジスタの間でのAND命令の場合と同じですから，そこを参照してください。

レジスタAと　あるメモリー間のAND命令

メモリーのアドレスを指定する方法が三つあります。それぞれ，記号，アセンブラ言語，機械語の順でまとめておきます。なお，IXやIYに加える値は，16進数23としてあります。

A←A∧M(HL)
AND　(HL)　……………A6

A←A∧M(IX＋23)
AND　(IX＋23H)
………　DD・A6・23

A←A∧M(IY＋23)
AND　(IY＋23H)
………　FD・A6・23

これらの命令が各フラグに与える影響は，レジスタAと他のレジスタの間のAND命令の場合と同じですから，そこを参照してください。

レジスタAと他のレジスタ間のOR命令

一例を記号で表わすと

A←A∨D

となります。今，レジスタA，Dの各ビットが次のようになっていたとします。

A＝01101011……（6B）

D＝10000001……（81）

ここで　A←A∨D　を実行すると，Aは

A＝11101011……（EB）

となります。レジスタA，Dでともにゼロになっているビットは第2ビットと第4ビットの二つです。したがって，この二つを除いたビットをすべて1にします。これを，レジスタAに入れるのですから，上のようになるわけです。OR命令はこのような命令なのです。

AND命令と同じように，入出装置に出し入れするコードの処理などによく用います。

この命令は全部で7個あり，それは次のようになります。

A←A∨B
OR　B　………………　B0

A←A∨C
OR　C　………………　B1

A←A∨D
OR　D　………………　B2

A←A∨E
OR　E　………………　B3

A←A∨H
OR　H　………………　B4

A←A∨L
OR　L　………………　B5

A←A∨A
OR　A　………………　B7

これらの命令が，各フラグに与える影響は，次の通りです。

S　：　この演算の結果，Aレジスタの第7ビット（最上位ビット）が1ならば，S＝1，そうでなければ，S＝0

Z　：　この演算の結果，Aレジスタのすべてのビットがゼロになった場合に，Z＝1，そうでなければ，Z＝0

P/V：　この演算の結果，Aレジスタの8個のビットの内，1になっているビットの数が，偶数ならば，P/V＝1，そうでなければ，P/V＝0

CY：　どのような場合も　CY＝0

レジスタAと直接数値とのORをとる命令

16進数2桁の数をXとすると，記号は次のようになります。

A←A∨Ẋ

アセンブラ言語では，

OR　X

です。機械語は

F6・X

です。Xが34ならば，アセンブラ言語，機械語は

OR　34H　……………　F6・34

となります。

この命令が各フラグに与える影響は，レジスタAと，他のレジスタの間でのORを求める場合と同じですので，そこを参照してください。

レジスタAとあるメモリーの間でORをとる命令

メモリーのアドレスを指定する方法のちがいによって，三つの種類があります。以下にまとめてみますが，IXやIYに加える数は，ここでは，16進数　ABとしておきます。

A←A∨M(HL)
OR　(HL)　……………　B6

A←A∨M(IX+AB)
OR　(IX+0ABH)
………DD・B6・AB

A←A∨M(IY+AB)
OR　(IY+0ABH)
………FD・B6・AB

これらの命令が，各フラグに与える影響は，レジスタAと他のレジスタのORをとる場合と同じですので，そこを参照してください。

レジスタAと他のレジスタの間の排他的論理和〔ExclusiveOr〕をとる命令

一例を記号で書くと

A←A∀B

となります。今，レジスタA，レジスタBが次のようなビットの立ち方をしている場合，

A＝10110000……(B0)

B＝01100000……(60)

このA←A∀B　を実行した後　Aは次のようになります。

A＝11010000……(D0)

この命令は，レジスタA，Bの対応するビットが，同じたち方をしている場合に，そのビットをゼロにして，他は1にするというものなのです。この場合，レジスタA，Bの第0，1，2，3ビットはともにゼロですし，第5ビットはともに1ですから，これらのビットがゼロになり，他は1になるわけです。したがって，レジスタAの内容は上に書いたようになるわけです。

この種の命令は7個あり，記号，アセンブラ言語，機械語の順でまとめると次のようになります。

A←A∀B
　XOR　B　……………　A8

B←A∀C
　XOR　C　……………　A9

A←A∀D
　XOR　D　……………　AA

A←A∀E
　XOR　E　……………　AB

A←A∀H
　XOR　H　……………　AC

A←A∀L
　XOR　L　……………　AD

A←A∀A
　XOR　A　……………　AF

レジスタAの内容をクリアしたいときに

XOR　A　　(A←A∀A)

をよく使います。

LD　A，0H　(A←0)

は2ワードにまたがる命令ですが，上の命令は1ワードですむからです。

さて，この命令が各フラグに与える影響についてまとめましょう。それは次のようになります。

S：　この演算の結果，レジスタAの最上位ビット（第7ビット）が1ならば，S＝1，そうでなければ，S＝0

Z：　この演算の結果，レジスタAのすべてのビットがゼロになれば，Z＝1，そうでなければ　Z＝0

P/V：　この演算の結果，レジスタAの8個のビットの内，1になったビットの数が偶数なら，P/V＝1，そうでなければ　P/V＝0

CY：　どのようた場合でも　CY＝0

レジスタAと直接数値との排他的論理和を求める命令

16進数2桁の数をXとすると，記号は次のようになります。

A←A∀X

アセンブラ言語の方は

XOR　X

です。機械語は

EE・X

です。XがFFなら，アセンブラ言語，機械語は次のようになります。

XOR　0FFH……………EE・FF

この命令が，各フラグに与える影響は，レジスタ間の場合と同じです　で，そこを参照してください。

レジスタAと，あるメモリーの間の排他的論理和をとる命令

メモリーのアドレスを指定する方法のちがいで，3種類の命令があります。以下まとめてみますがIXやIYに加える数値は，16進数，78としておきます。

A←A∀M(HL)
　XOR　(HL)　…………　AE

A←A∀M(IX＋78)

XOR　(IX+78H)
…………DD・AE・78

A←A∀M(IY+78)
XOR　(IY+78H)
…………FD・AE・78

これらの命令が，各フラグに与える影響は，レジスタ間で，排他的論理和をとる場合と同じですから，そこを参照してください。

AND命令，OR命令，XOR命令のテストプログラムの作成

AND，OR，XORの各命令で，レジスタAのビットが，どのように変わるのかを見るために**第3－4図**に示したフローチャートを考えてみました。**第3－4図**を見ればわかると思いますが，プログラムがうまく働けば，実行後

M(ADR1)=83
M(ADR2)=E7
M(ADR3)=18

となるはずです。

第3－5図のように，アセンブラプログラミングして実際のアドレスを割りつけてみると

ADR1=7D12
ADR2=7D13
ADR3=7D14

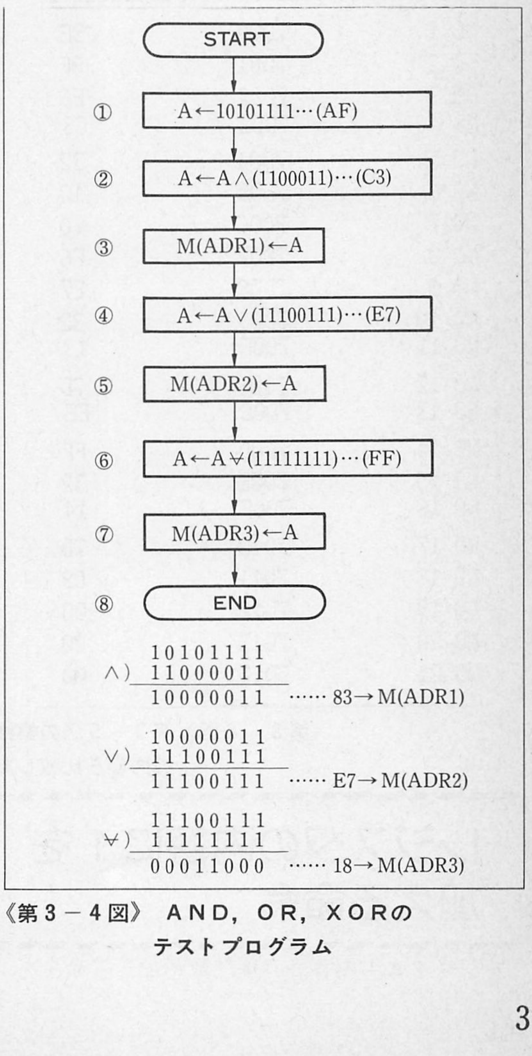
であることがわかります。

これらをもとに，機械語をメモリーに入れ，実行前と実行後のメモリーの内容を比較したものが，**第3－5図**です。

予定通り

M(7D12)=83
M(7D13)=E7
M(7D14)=18

になっていることがわかります。



《第3－4図》 AND，OR，XORのテストプログラム

```
MACRO-80 3.3    09-MAY-79          PAGE    1
                              00100  ;          TEST PROGRAM ( AND, OR, XOR )
                              00200  .Z80
0000'                         00300  ASEG
                              00400  ORG        7D00H
                              00500  ;
7D00     3E AF                00600             LD      A,10101111B      ; A=AF (BINARY 10101111)
7D02     E6 C3                00700             AND     11000011B        ; A=( A AND 11000011 )
7D04     32 7D12              00800             LD      (ADR1),A         ; M(ADR1)=A .
7D07     F6 E7                00900             OR      11100111B        ; A=( A OR 11100111 )
7D09     32 7D13              01000             LD      (ADR2),A         ; M(ADR2)=A
7D0C     EE FF                01100             XOR     11111111B        ; A=( A XOR 11111111 )
7D0E     32 7D14              01200             LD      (ADR3),A         ; M(ADR3)=A
7D11     C9                   01300             RET                      ; RETURN
7D12     00                   01400  ADR1:      DB      00H              ; M(ADR1)=0
7D13     00                   01500  ADR2:      DB      00H              ; M(ADR2)=0
7D14     00                   01600  ADR3:      DB      00H              ; M(ADR3)=0
                              01700             END
```

《第３－５図》第３－４図のフローチャートをアセンブラプログラミングしたもの

** BEFORE RUN

NO 1	7D00	3E
NO 2	7D01	AF
NO 3	7D02	E6
NO 4	7D03	C3
NO 5	7D04	32
NO 6	7D05	12
NO 7	7D06	7D
NO 8	7D07	F6
NO 9	7D08	E7
NO 10	7D09	32
NO 11	7D0A	13
NO 12	7D0B	7D
NO 13	7D0C	EE
NO 14	7D0D	FF
NO 15	7D0E	32
NO 16	7D0F	14
NO 17	7D10	7D
NO 18	7D11	C9
NO 19	7D12	00
NO 20	7D13	00
NO 21	7D14	00

** AFFTER RUN

NO 1	7D00	3E
NO 2	7D01	AF
NO 3	7D02	E6
NO 4	7D03	C3
NO 5	7D04	32
NO 6	7D05	12
NO 7	7D06	7D
NO 8	7D07	F6
NO 9	7D08	E7
NO 10	7D09	32
NO 11	7D0A	13
NO 12	7D0B	7D
NO 13	7D0C	EE
NO 14	7D0D	FF
NO 15	7D0E	32
NO 16	7D0F	14
NO 17	7D10	7D
NO 18	7D11	C9
NO 19	7D12	83
NO 20	7D13	E7
NO 21	7D14	18

《第３－６図》第３－５図の機械語をメモリーに入れ、実行前と実行後のメモリーの内容を比較したもの

レジスタの内容に１を加える命令

レジスタＡの内容に直接，数を加える命令を，前にお話ししました。その命令があれば，ここでとりあげる命令は不必要なのではないか，と思われるかもしれません。しかし，レジスタの内容に１を加えるという命令は，意外と多く使うのです。それがどのレジスタであっても１ワード命令でか

たずくのは，かなり助かるのです。それと，もう一つ，レジスタＡの内容に数値を加えるのとは，ちがう結果をもたらすことがあります。それは，この命令が，キャリーフラグに影響を及ぼさない点です。この点には注意しておいてください。

この種の命令は７個あります。下にまとめてみましょう。

Ｂ←Ｂ＋１
　ＩＮＣ　Ｂ　…………………０４

Ｃ←Ｃ＋１
　ＩＮＣ　Ｃ　…………………０Ｃ

Ｄ←Ｄ＋１
　ＩＮＣ　Ｄ　…………………１４

Ｅ←Ｅ＋１
　ＩＮＣ　Ｅ　…………………１Ｃ

Ｈ←Ｈ＋１
　ＩＮＣ　Ｈ　…………………２４

Ｌ←Ｌ＋１
　ＩＮＣ　Ｌ　…………………２Ｃ

Ａ←Ａ＋１
　ＩＮＣ　Ａ　…………………３Ｃ

これらの命令が，各種フラグに与える影響を次にまとめておきましょう。

Ｓ：　演算の結果，指定したレジスタの最上位のビットが１であれば，Ｓ＝１，そうでなければ，Ｓ＝０

Ｚ：　演算の結果，指定したレジスタのすべてのビットがゼロならば，　Ｚ＝１，そうでなければ，Ｚ＝０

P/V：　演算を実行する前に，指定したレジスタが16進数　７Ｆ　であった場合に，P/V＝１，そうでなければ，P/V＝０

ＣＹ：　影響をうけない（つまり，前の値のまま）

あるメモリーの内容に１を加える命令

メモリーのアドレスを指定する方法が三つあります。ＨＬレジスタを使うやり方，インデックスレジスタＩＸ，ＩＹを使うやり方です。記号，アセンブラ言語，機械語の順にまとめると，次のようになります。

Ｍ(ＨＬ)←Ｍ(ＨＬ)＋１
　ＩＮＣ　(ＨＬ)　…………３４

Ｍ(ＩＸ＋０ＡＢ)←Ｍ(ＩＸ＋０ＡＢ)＋１
　ＩＮＣ　(ＩＸ＋０ＡＢＨ)
　　　…………ＤＤ・３４・ＡＢ

Ｍ(ＩＹ＋０ＡＢ)←Ｍ(ＩＸ＋０ＡＢ)＋１
　ＩＮＣ　(ＩＹ＋０ＡＢＨ)
　　　…………ＦＤ・３４・ＡＢ

ただし，インデックスレジスタＩＸ，ＩＹに加える数値を，上の場合，16進数，ＡＢにしてあります。

これらの命令が，各フラグに与える影響は，レジスタの内容に１を加える命令の場合と同じです。そこで，〝指定したレジスタ〟とあるところを，〝指定したメモリー〟と直せばよいわけです。

レジスタの内容から１を引く命令

各レジスタの内容から１を引く命令が用意されています。記号，アセンブラ言語，機械語の順にまとめてみましょう。

Ｂ←Ｂ－１
　ＤＥＣ　Ｂ　…………０５

Ｃ←Ｃ－１
　ＤＥＣ　Ｃ　…………０Ｄ

Ｄ←Ｄ－１
　ＤＥＣ　Ｄ　…………１５

Ｅ←Ｅ－１
　ＤＥＣ　Ｅ　…………１Ｄ

Ｈ←Ｈ－１
　ＤＥＣ　Ｈ　…………２５

Ｌ←Ｌ－１
　ＤＥＣ　Ｌ　…………２Ｄ

Ａ←Ａ－１
　ＤＥＣ　Ａ　…………３Ｄ

これらの命令が，各フラグに与える影響は次の通りです。P/Vと，特にＣＹには注意してください。

Ｓ：　演算の後，指定したレジスタの第７ビット（最上位ビット）が１であればＳ＝１，そうでなければ，Ｓ＝０

Ｚ：　演算の後，指定したレジスタのすべ

てのビットかゼロならば，Z＝1，そうでなければ，Z＝0

P/V：　演算を行なう前に，指定したレジスタの内容が16進で80ならば，P/V＝1，そうでなければ，P/V＝0

CY：　どのような場合も影響をうけない。つまり，前の値を保つ

あるメモリーの内容から1を引く命令

メモリーのアドレスを指定する方法が三つあります。HLレジスタを用いる場合と，インデックスレジスタIX，IYを用いる場合とです。インデックスレジスタを使う場合には，何も加えないものとして，下に，記号，アセンブラ言語，機械語の順でまとめてみましょう。

M(HL)←M(HL)－1
　DEC　(HL)　…………35

M(IX)←M(IX)－1
　DEC　(IX＋0H)
　…………DD・35・00

M(IX)←M(IY)－1
　DEC　(IY＋0H)
　…………FD・35・00

これらの命令が，各フラグに与える影響は，レジスタの内容から1を引く場合と同じです。そこの〝指定したレジスタの〟の文を〝指定したメモリーの〟と読みかえて，理解してください。

INC,DEC命令を使ったテストプログラムの作成

INC，DEC命令は特にむずかしい命令ではありませんが，一応テストプログラムを作ってみました。**第3－7図**がそのフローチャートです。HLレジスタのLから1ずつ引いて，レジスタAからレジスタEまでの内容を入れるメモリーのアドレスを1ずつ小さくしています。

・レジスタAからレジスタEまでは，1から1ずつ大きな数が作られていきます。

実行後は，

M(ADR－4)＝5
M(ADR－3)＝4
M(ADR－2)＝3
M(ADR－1)＝2
M(ADR)＝1

となるはずです。

第3－8図は，**第3－7図**のフローチャートをもとに，アセンブラプログラミングしたものです。注意すべき点は，アドレス　7D02　の

7D02　21　7D1B

をメモリーに書き込む場合，実際には

21・1B・7D

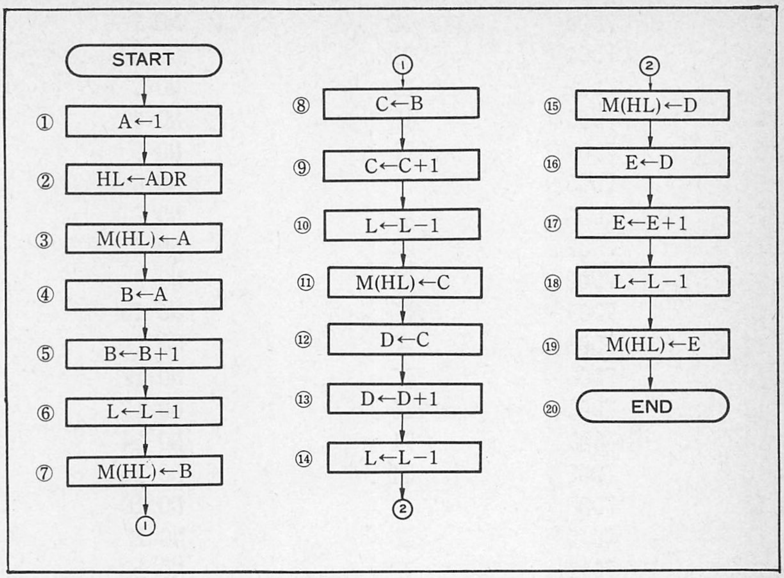
とする点です。

それと，シンボル，ADR　の使い方に注意してください。

DBは，ディファイン・バイトで，直接メモリーに数値を書き込むときに用いるものです。

第3－9図は，**第3－8図**から得られた機械語を，直接メモリーに書き込み，このプログラムを

実行する前と，実行後のメモリーの内容を比較し　　　たものです。



《第３－７図》
ＩＮＣ、ＤＥＣ命令のテスト プログラム

```
MACRO-80 3.3    09-MAY-79        PAGE    1
                         00100  ;          TEST PROGRAM ( INC, DEC )
                         00200  .Z80
0000'                    00300  ASEG
                         00400  ORG        7D00H
                         00500  ;
7D00    3E 01            00600             LD      A,1H        ; A=1
7D02    21 7D1B          00700             LD      HL,ADR      ; HL=ADR
7D05    77               00800             LD      (HL),A      ; M(ADR)=A
7D06    47               00900             LD      B,A         ; B=A
7D07    04               01000             INC     B           ; B=B+1
7D08    2D               01100             DEC     L           ; L=L-1
7D09    70               01200             LD      (HL),B      ; M(ADR-1)=B
7D0A    48               01300             LD      C,B         ; C=B
7D0B    0C               01400             INC     C           ; C=C+1
7D0C    2D               01500             DEC     L           ; L=L-1
7D0D    71               01600             LD      (HL),C      ; M(ADR-2)=C
7D0E    51               01700             LD      D,C         ; D=C
7D0F    14               01800             INC     D           ; D=D+1
7D10    2D               01900             DEC     L           ; L=L-1
7D11    72               02000             LD      (HL),D      ; M(ADR-3)=D
7D12    5A               02100             LD      E,D         ; E=D
7D13    1C               02200             INC     E           ; E=E+1
7D14    2D               02300             DEC     L           ; L=L-1
7D15    73               02400             LD      (HL),E      ; M(ADR-4)=E
7D16    C9               02500             RET                 ; RETURN
7D17    00               02600             DB      00H         ; M(ADR-4)=0
7D18    00               02700             DB      00H         ; M(ADR-3)=0
7D19    00               02800             DB      00H         ; M(ADR-2)=0
7D1A    00               02900             DB      00H         ; M(ADR-1)=0
7D1B    00               03000  ADR        DB      00H         ; M(ADR)=0
                         03100             END
```

《第３－８図》第３－７図のフローチャートをアセンブラプログラミングしたもの

** BEFORE RUN		
NO 1	7D00	3E
NO 2	7D01	01
NO 3	7D02	21
NO 4	7D03	1B
NO 5	7D04	7D
NO 6	7D05	77
NO 7	7D06	47
NO 8	7D07	04
NO 9	7D08	2D
NO 10	7D09	70
NO 11	7D0A	48
NO 12	7D0B	0C
NO 13	7D0C	2D
NO 14	7D0D	71
NO 15	7D0E	51
NO 16	7D0F	14
NO 17	7D10	2D
NO 18	7D11	72
NO 19	7D12	5A
NO 20	7D13	1C
NO 21	7D14	2D
NO 22	7D15	73
NO 23	7D16	C9
NO 24	7D17	00
NO 25	7D18	00
NO 26	7D19	00
NO 27	7D1A	00
NO 28	7D1B	00

** AFFTER RUN		
NO 1	7D00	3E
NO 2	7D01	01
NO 3	7D02	21
NO 4	7D03	1B
NO 5	7D04	7D
NO 6	7D05	77
NO 7	7D06	47
NO 8	7D07	04
NO 9	7D08	2D
NO 10	7D09	70
NO 11	7D0A	48
NO 12	7D0B	0C
NO 13	7D0C	2D
NO 14	7D0D	71
NO 15	7D0E	51
NO 16	7D0F	14
NO 17	7D10	2D
NO 18	7D11	72
NO 19	7D12	5A
NO 20	7D13	1C
NO 21	7D14	2D
NO 22	7D15	73
NO 23	7D16	C9
NO 24	7D17	05
NO 25	7D18	04
NO 26	7D19	03
NO 27	7D1A	02
NO 28	7D1B	01

《3－9図》 第3－8図のプログラムをもとに，メモリーに機械語を入れ，実行前と実行後のメモリーの内容を比較したもの

4 16ビットの演算命令

8ビットの演算命令ほど多くはありませんが，アドレスを指定したり，変化させたりする場合に必要なものを中心にして，いくらかの命令があります。

レジスタペアどうしの足し算

一例を記号で表わしてみましょう。

HL←HL＋DE

16進4桁の数どうしの足し算がこれでできます。

同じような命令は，全部で12個あります。次に記号，アセンブラ言語，機械語の順にまとめてみましょう。

ＨＬ←ＨＬ＋ＢＣ
　ＡＤＤ　ＨＬ，ＢＣ　………………０９

ＨＬ←ＨＬ＋ＤＥ
　ＡＤＤ　ＨＬ，ＤＥ　………………１９

ＨＬ←ＨＬ＋ＨＬ
　ＡＤＤ　ＨＬ，ＨＬ　………………２９

ＨＬ←ＨＬ＋ＳＰ
　ＡＤＤ　ＨＬ，ＳＰ　………………３９

ＩＸ←ＩＸ＋ＢＣ
　ＡＤＤ　ＩＸ，ＢＣ　………ＤＤ・０９

ＩＸ←ＩＸ＋ＤＥ
　ＡＤＤ　ＩＸ，ＤＥ　………ＤＤ・19

ＩＸ←ＩＸ＋ＩＸ
　ＡＤＤ　ＩＸ，ＩＸ　………ＤＤ・２９

ＩＸ←ＩＸ＋ＳＰ
　ＡＤＤ　ＩＸ，ＳＰ　………ＤＤ・３９

ＩＹ←ＩＹ＋ＢＣ
　ＡＤＤ　ＩＹ，ＢＣ　………ＦＤ・０９

ＩＹ←ＩＹ＋ＤＥ
　ＡＤＤ　ＩＹ，ＤＥ　………ＦＤ・１９

ＩＹ←ＩＹ＋ＩＹ
　ＡＤＤ　ＩＹ，ＩＹ　………ＦＤ・２９

ＩＹ←ＩＹ＋ＳＰ
　ＡＤＤ　ＩＹ，ＳＰ　………ＦＤ・３９

これらの命令で，特に注意しなければいけない点があります。フラグへの影響です。8ビットの場合とは異なるからです。以下，それをまとめてみましょう。

S　：　どんな場合にも影響をうけず，前の値を保持する

Z　：　どんな場合にも影響をうけず，前の

値を保持する

P/V：　どんな場合にも影響をうけず，前の値を保持する

CY：　演算の結果が，１６ビットで表わせなかった場合に，ＣＹ＝１，そうでなければ，ＣＹ＝０

レジスタペアどうしの和にキャリーの値を加える場合の命令

これは，３ワード，あるいは４ワード以上にまたがる数値の和を求める場合や，正負を考慮した上で１６ビットどうしの和を求める場合にかなりの力を発揮します。考え方は，１ワード単位でのＡＤＣ命令と同じです。

これらの命令は，全部で４個あります。記号，アセンブラ言語，機械語の順でまとめると，次のようになります。

```
HL←HL+BC+CY
   ADC  HL, BC  ………ED・4A
HL←HL+DE+CY
   ADC  HL, DE  ………ED・5A
HL←HL+HL+CY
   ADC  HL, HL  ………ED・6A
HL←HL+SP+CY
   ADC  HL, SP  ………ED・7A
```

これらの命令が各フラグに与える影響は，先のレジスタペアの場合のＡＤＤ命令とは異なり，８ビットの場合のＡＤＣ，またはＡＤＤ命令の場合と同じです。以下まとめてみましょう。

S　：演算の後，レジスタペアＨＬの最上位ビット（レジスタＨの第７ビット）が１ならば，Ｓ＝１，そうでなければＳ＝０

Z　：　演算の後，ＨＬレジスタの16個のビットすべてがゼロの場合，Ｚ＝１，そうでなければ，Ｚ＝０

P/V：　16ビットで表わされた正負の数の演算において，実際表示との正負が異なってしまう場合に，P/V＝１，そうでない場合にP/V＝０

CY：　演算の結果が，16ビットを越えてしまう場合，越えた分をＣＹが表す

レジスタペアどうしの差からキャリーの値を引く場合の命令

この命令は，３ワード，あるいは４ワード以上にまたがる数値どうしの差を求める場合や，正負を考慮した上での16ビットどうしの差を求める場合に，かなりの力を発揮する命令です。

全部で４種類あります。以下，記号，アセンブラ言語，機械語の順でまとめてみましょう。

```
HL←HL-BC-CY
   SBC  HL, BC…………ED・42
HL←HL-DE-CY
   SBC  HL, DE…………ED・52
HL←HL-HL-CY
   SBC  HL, HL…………ED  62
HL←HL-SP-CY
   SBC  HL, SP…………ED・72
```

この命令が，各フラグに与える影響は，１ワード単位のＳＢＣ命令と同じです。以下まとめておきましょう。

S　：　演算の後，レジスタペアＨＬの最上位ビット（レジスタＨの第７ビット）が１ならば，Ｓ＝１，そうでなければＳ＝０

Z　：　演算の後，レジスタペアＨＬの16個のビットすべてがゼロならば，Ｚ＝１，そうでなければ，Ｚ＝０

P/V：　16ビットで正負を考慮した上で，引き算をした場合，正負，それぞれの場合でオーバーフローを起こしたときにP/V＝１，そうでなければ P/V＝０

CY：　二つの16ビットの数が，すべて正の数を表わしているとみなした場合，引く方の数がＨＬの内容より大きい場合に，ＣＹ＝１，そうでない場合にＣＹ＝０

レジスタペアの内容に1を加える命令

指定するアドレスを1ずつ変えていくときに便利です。また，この命令は，各種フラグに全く影響を与えません。

全部で6種類の命令があります。以下，記号，アセンブラ言語，機械語の順でまとめてみましょう。

```
BC←BC+1
    INC  BC  ………………03
DE←DE+1
    INC  DE  ………………13
HL←HL+1
    INC  HL  ………………23
SP←SP+1
    INC  SP  ………………33
IX←IX+1
    INC  IX  ………………DD・23
IY←IY+1
    INC  IY  ………………FD・23
```

レジスタペアの内容から1を引く命令

この命令は，指定するアドレスを1ずつ変えていく場合に便利です。各種フラグにも影響を与えないので，使い方によっては，かなり便利です。

全部で6個の命令がありますが，下に記号，アセンブラ言語，機械語の順でまとめておきました。

```
BC←BC−1
    DEC  BC  ………………0B
DE←DE−1
    DEC  DE  ………………1B
HL←HL−1
    DEC  HL  ………………2B
SP←SP−1
    DEC  SP  ………………3B
IX←IX−1
    DEC  IX  ………………DD・2B
IY←IY−1
    DEC  IY  ………………FD・2B
```

レジスタペアにおけるADC，SBC命令を使ったテストプログラムの作成

第4−1図にそのフローチャートを示しておきましたが，4ワードにまたがる数値どうしの和と差を求めてみましょう。考え方は，1ワード単位におけるADC，SBC命令のところでお話しした，2ワードにまたがる数値どうしの和と差の求め方の場合と同じです。

ここで注意すべき点は，Z80に

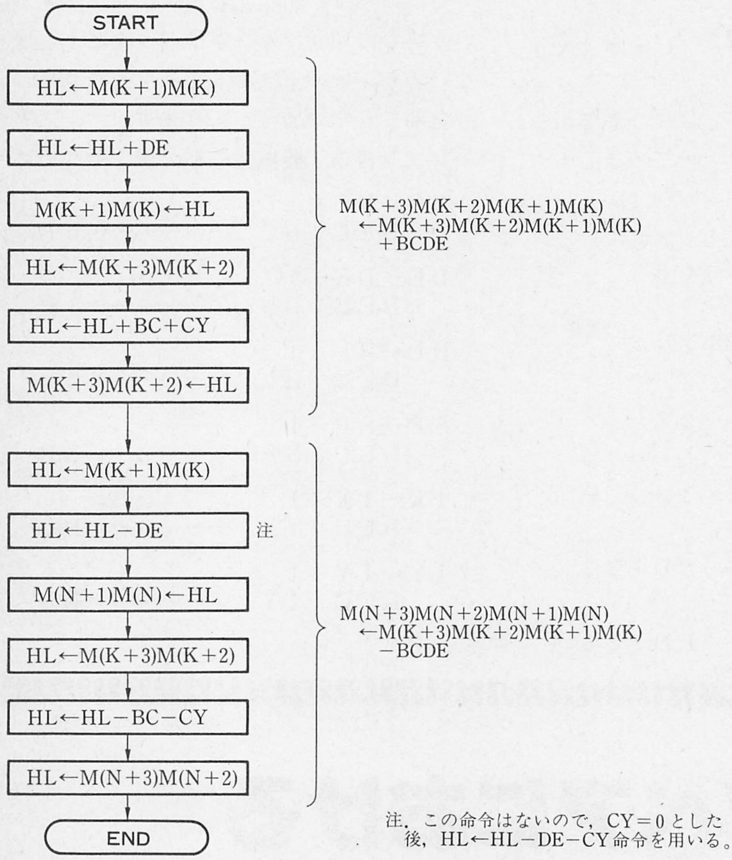
HL←HL−DE

という命令がない，という点です。したがって，実際には，キャリーをゼロにした後で

HL←HL−DE−CY

を実行することになります。キャリーをゼロにする方法は，何通りかあります。たとえば，論理演算の命令を実行すると，キャリーはどんな場合で

M(K+3)M(K+2)M(K+1)M(K)
←M(K+3)M(K+2)M(K+1)M(K)+BCDE
M(N+3)M(N+2)M(N+1)M(N)
←M(K+3)M(K+2)M(K+1)M(K)−BCDE



《第４－１図》
レジスタペアにおける
ＡＤＣ，ＳＢＣ命令を使
ったテストプログラム

もゼロになることを前にお話ししました。

そこで，ここでは

Ａ←Ａ ∀ Ａ

を使ってみることにしましょう。

さて，ＢＣＤＥは

ＢＣＤＥ＝０ＥＦＦＦＦＦＦＦ

メモリーＭ(Ｋ＋３)Ｍ(Ｋ＋２)Ｍ(Ｋ＋１)

Ｍ(Ｋ)には０００００００１

が入るようにして，アセンブラプログラミングしてみました。それが**第４－２図**です。

ここでは，シンボルＫ，Ｎの使い方，また，Ｋ，Ｎを定義した場合にＫ＋２，Ｎ＋２などの使い方のできる点に注意してください。

つまり，シンボルＫはアドレス７Ｄ27に，シンボルＮはアドレス７Ｄ２Ｂに対応しています。このとき，アセンブラ言語を使用中にＫ＋２を使えば，これは７Ｄ27に２を加えたときの値，７Ｄ29に対応するという点です。同じようにして，Ｎ＋２は７Ｄ２Ｄに対応しています。

```
    ０ＥＦＦＦＦＦＦＦ
＋)  ０００００００１
―――――――――――――
    Ｆ０００００００
```

ですから，プログラム実行後は，

Ｍ(７Ｄ２７)＝００

Ｍ(７Ｄ２８)＝００

Ｍ(７Ｄ２９)＝００

Ｍ(７Ｄ２Ａ)＝Ｆ０

となるはずですし，引き算のほうでは

```
    ０Ｆ０００００００
－)  ０ＥＦＦＦＦＦＦＦ
―――――――――――――
    ０００００００１
```

ですから

```
   MACRO-80 3.3    09-MAY-79       PAGE    1
                               00100  ;         TEST PROGRAM ( ADC HL, SBC HL )
                               00200  .Z80
 0000'                         00300  ASEG
                               00400  ORG       7D00H
                               00500  ;
 7D00     01 EFFF              00600            LD        BC,0EFFFH       ; BC=EFFF
 7D03     11 FFFF              00700            LD        DE,0FFFFH       ; DE=FFFF
 7D06     2A 7D27              00800            LD        HL,(K)          ; HL=M(K+1)M(K)
 7D09     19                   00900            ADD       HL,DE           ; HL=HL+DE
 7D0A     22 7D27              01000            LD        (K),HL          ; M(K+1)M(K)=HL
 7D0D     2A 7D29              01100            LD        HL,(K+2)        ; HL=M(K+3)M(K+2)
 7D10     ED 4A                01200            ADC       HL,BC           ; HL=HL+BC+CY
 7D12     22 7D29              01300            LD        (K+2),HL        ; M(K+3)M(K+2)=HL
                               01400  ;
 7D15     2A 7D27              01500            LD        HL,(K)          ; HL=M(K+1)M(K)
 7D18     AF                   01600            XOR       A               ; CY=0
 7D19     ED 52                01700            SBC       HL,DE           ; HL=HL-DE-CY
 7D1B     22 7D2B              01800            LD        (N),HL          ; M(N+1)M(N)=HL
 7D1E     2A 7D29              01900            LD        HL,(K+2)        ; HL=M(K+3)M(K+2)
 7D21     ED 42                02000            SBC       HL,BC           ; HL=HL-BC-CY
 7D23     22 7D2D              02100            LD        (N+2),HL        ; M(N+3)M(N+2)=HL
 7D26     C9                   02200            RET                       ; RETURN
 7D27     01                   02300  K:        DB        01H             ; M(K)=01
 7D28     00                   02400            DB        00H             ; M(K+1)=00
 7D29     00                   02500            DB        00H             ; M(K+2)=00
 7D2A     00                   02600            DB        00H             ; M(K+3)=00
 7D2B     00                   02700  N:        DB        00H             ; M(N)=00
 7D2C     00                   02800            DB        00H             ; M(N+1)=00
 7D2D     00                   02900            DB        00H             ; M(N+2)=00
 7D2E     00                   03000            DB        00H             ; M(N+3)=00
                               03100            END
```

《第４－２図》第４－１図のフローチャートをもとにアセンブラプログラミングしたもの

M(７D２B)＝０１
M(７D２C)＝００
M(７D２D)＝００
M(７D２E)＝００

となるはずです。

機械語を直接メモリーに書き込み，実行前と実行後のメモリーの内容の比較を行なったものが**第４－３図**，**第４－４図**です。期待どうりの値になっています。

** BEFORE RUN

NO 1	7D00	01
NO 2	7D01	FF
NO 3	7D02	EF
NO 4	7D03	11
NO 5	7D04	FF
NO 6	7D05	FF
NO 7	7D06	2A
NO 8	7D07	27
NO 9	7D08	7D
NO 10	7D09	19
NO 11	7D0A	22
NO 12	7D0B	27
NO 13	7D0C	7D
NO 14	7D0D	2A
NO 15	7D0E	29
NO 16	7D0F	7D
NO 17	7D10	ED
NO 18	7D11	4A
NO 19	7D12	22
NO 20	7D13	29
NO 21	7D14	7D
NO 22	7D15	2A
NO 23	7D16	27
NO 24	7D17	7D
NO 25	7D18	AF
NO 26	7D19	ED
NO 27	7D1A	52
NO 28	7D1B	22
NO 29	7D1C	2B
NO 30	7D1D	7D
NO 31	7D1E	2A
NO 32	7D1F	29
NO 33	7D20	7D
NO 34	7D21	ED
NO 35	7D22	42
NO 36	7D23	22
NO 37	7D24	2D
NO 38	7D25	7D
NO 39	7D26	C9
NO 40	7D27	01
NO 41	7D28	00
NO 42	7D29	00
NO 43	7D2A	00
NO 44	7D2B	00
NO 45	7D2C	00
NO 46	7D2D	00
NO 47	7D2E	00

《第４－３図》第４－２図のプログラムを直接メモリーに書き込んだもの

** AFFTER RUN

NO 1	7D00	01
NO 2	7D01	FF
NO 3	7D02	EF
NO 4	7D03	11
NO 5	7D04	FF
NO 6	7D05	FF
NO 7	7D06	2A
NO 8	7D07	27
NO 9	7D08	7D
NO 10	7D09	19
NO 11	7D0A	22
NO 12	7D0B	27
NO 13	7D0C	7D
NO 14	7D0D	2A
NO 15	7D0E	29
NO 16	7D0F	7D
NO 17	7D10	ED
NO 18	7D11	4A
NO 19	7D12	22
NO 20	7D13	29
NO 21	7D14	7D
NO 22	7D15	2A
NO 23	7D16	27
NO 24	7D17	7D
NO 25	7D18	AF
NO 26	7D19	ED
NO 27	7D1A	52
NO 28	7D1B	22
NO 29	7D1C	2B
NO 30	7D1D	7D
NO 31	7D1E	2A
NO 32	7D1F	29
NO 33	7D20	7D
NO 34	7D21	ED
NO 35	7D22	42
NO 36	7D23	22
NO 37	7D24	2D
NO 38	7D25	7D
NO 39	7D26	C9
NO 40	7D27	00
NO 41	7D28	00
NO 42	7D29	00
NO 43	7D2A	F0
NO 44	7D2B	01
NO 45	7D2C	00
NO 46	7D2D	00
NO 47	7D2E	00

《第４－４図》第４－３図のプログラムを実行した後のメモリーの内容

5 比較命令と条件ジャンプ命令

ここでは，レジスタAの値と，他のレジスタ，またはメモリー内の値との大小関係等の比較を行なう命令と，各フラグの状態を調べ，それに従ってジャンプする命令とをとりあげます。

比較命令

二つの数を比較するてっとりばやい方法は，その差をとってみることです。

しかし，その差そのものは必要ではありません。その差を求める段階で変化したフラグの値だけわかればよいのです。そこで，差を求めてはみるが，その差は無効にする，という命令を作り，それを比較命令と呼んでいます。

一例を記号で表わすと

Null ← A − B

または，単に

A − B

と書くこともあります。この命令は，結果を無効(null)にすること以外は，SUB命令と全く同じです。

レジスタ間，レジスタと直接数値との間，レジスタとメモリーの間，すべてあわせて11個の命令が用意されています。直接数値の場合は，16進数7Eとして，インデックスレジスタを用いる場合は，そこに加える値を16進数12として，以下これら11個の命令を記号，アセンブラ言語，機械語の順でまとめておきます。

Null←A－B
　CP　B　……………………B8

Null←A－C
　CP　C　……………………B9

Null←A－D
　CP　D　……………………BA

Null←A－E
　CP　E　……………………BB

Null←A－H
　CP　H　……………………BC

Null←A－L
　CP　L　……………………BD

Null←A－A
　CP　A　……………………BF

Null←A－7E

```
CP  7EH  ………………FE・7E
Null←M (HL)
CP  (HL)  ……………BE
Null←M (IX+12)
CP  (IX+12H)…DD・BE・12
Null←M (IY+12)
CP  (IY+12H)…FD・BE・12
```

これらの命令が，各フラグに与える影響は次の通りです。

S ：差を求めてみたとき，その最上位のビットが1ならば，S＝1，そうでなければ，S＝0

Z ：二つの数の間に差がないとき，Z＝1，そうでなければZ＝0

P/V：8ビット内で正負を表わすとき，正負それぞれの間でオーバーフローを起こせばP/V＝1，そうでなければP/V=0

CY：8ビットがすべて正の数を表わしているとみなしたとき，引いてみる方の数が引かれる方の数より小さければ，CY＝1，そうでなければ，CY＝0

無条件ジャンプ命令

プログラムは一般にアドレスの若い方から順に実行されますが，その流れを変えたいときに用いる命令が，無条件ジャンプ命令です。この命令が実行されればプログラマーが指定する任意のアドレスへジャンプします。

そのジャンプ先のアドレスを指定する方法が5通りあります。

①直接，ジャンプ先のアドレスを指定する方法
②その命令を実行しているときのプログラムカウンタに，正負を考慮した2桁の16進数を加えることによってジャンプする方法
③レジスタペアHLに書かれている内容を，プログラムカウンタに送ることによってジャンプさせる方法
④インデックスレジスタIXに書かれている内容を，プログラムカウンタに送ることによってジャンプさせる方法
⑤インデックスレジスタIYに書かれている内容を，プログラムカウンタに送ることによってジャンプさせる方法

①②については，後でお話しする条件ジャンプ命令に共通した点が多いので，特に注意してください。

それでは一つ一つ説明していきましょう。

①は機械語の3ワード目，2ワード目に直接，ジャンプ先のアドレスを書き込む方法です。したがって，ジャンプ先に制限はありませんが，この命令を使って作ったプログラムは，そのまま他のアドレスに書き込んで実行することはできません。

ジャンプ先のアドレスがABCDで，そこにシンボルSYMがつけられている場合，アセンブラ言語は次のようになります。

```
JP  0ABCDH
```

または

```
JP  SYM
```

機械語は

```
C3・CD・AB
```

となります。

②は機械語の2ワード目に，この命令が実行されている時点のプログラムカウンタに加える数を書き込む方法です。

プログラムカウンタは一つの命令が実行されているときに，次の命令の書かれている先頭のアドレスが入っています。このジャンプ命令は2ワード命令ですから，これが7D00から書かれていれば，この命令を実行中，プログラムカウンタの値は7D02となっています。この命令の2ワード目に10という値が入っていれば，

7D02＋10＝7D12

より，プログラムカウンタは7D12となります。したがって，この命令が終わりしだい，7D12番地へジャンプすることになります。

この命令の場合，プログラムカウンタに加えられる数が8ビットで表わせる正負の数ということから，ジャンプ先には制限があります。つまりプログラムカウンタには10進数で－128から＋127までの数しか加えることができないので，マイナス方向には126，プラス方向には129ワード分しかジャンプできません。

しかし，一つの独立したプログラム中で，ジャ

ンプ命令等がこのような方法を使っていれば，そのプログラムは，どこのメモリーに書かれようとも所定の働きをします。こういったプログラムをリロケイタブルであるといいます。

7D00にこの命令を書き，7D12にジャンプさせる場合，7D12にSYMというシンボルがついていれば，アセンブラ言語は，

JR　SYM

または，

JR　7D12H

とします。このときの機械語は

18・10

となります。

この命令を7D7F番地から書き，7D23番地へジャンプさせる場合の，機械語を考えてみましょう。

まず，7D7F，7D23の下2桁をとり，ジャンプ先のアドレスから，命令を書くアドレスの値を引く式を作ります。

23－7F

それぞれの桁を10進に直して

（02・03）－（07・15）

とします。　03－15　において，引く方の数が大きいので，上の桁から1桁分，つまり16を借りてきます。式は次のようになります。

（01・19）－（07・15）

これで，下の桁は計算ができるようになりましたが，上の桁においても，引く数の方が大きいので01の方に16を加えます。

（17・19）－（07・15）

となります。これを計算すると，

（10・04）

となります。これから，さらに2を引きます。

（10・04）－（00・02）＝（10・02）

となりました。これを16進数になおすと，

A2

となります。

したがって，求める機械語は，

18・A2

ということになります。

③④⑤は　ジャンプ先のアドレスをHL,IX,IYに書き込んでおいて，それらの命令を実行すればよいので，それほど問題はないと思います。

プログラム中で色々な計算をし，その値によって，ジャンプ先が変わる，というときに，この命令は便利です。

アセンブラ言語，機械語の順でまとめてみましょう。

③　JP　（HL）　…………………　E9
④　JP　（IX）　…………　DD・E9
⑤　JP　（IY）　…………　FD・E9

以上，五つの命令は，いずれも各フラグに影響を与えません。

ゼロブランチ命令

Zフラグを調べ，それが1ならば，2ワード目以降で指定するアドレスへジャンプする命令です。

これも2通りのやり方があります。別々にまとめてみましょう。

1，直接ジャンプ先のアドレスを指定する場合

ジャンプ先にはシンボルがつけられ，それがSYMである場合，アセンブラ言語は

JP　Z，SYM

ジャンプ先にシンボルがついていない場合，そのアドレスが　7D12なら，

JP　Z，7D12H

この場合の機械語は

CA・12・7D

となります。

2．命令の書かれているアドレスとジャンプ先のアドレスとの隔りを指定することによって，ジャンプ先のアドレスを割り出す場合

ジャンプ先にシンボルがつけられ，それがSYMである場合，アセンブラ言語は，

JR　Z，SYM

この命令が書かれている先頭アドレスが7D00で，ジャンプ先が7D12なら

JR　Z，7D12H

でも大丈夫ですし，そのときの機械語は

28・10

となります。2ワード目の10の計算方法は，無条件ジャンプのところを参照してください。

これらの命令は，各フラグに影響は与えません。

非ゼロブランチ命令

Zフラグを調べ，それがゼロならば，機械語の2ワード目以降で指定するアドレスへジャンプする命令です。

2通りの命令があります。

1．直接ジャンプ先のアドレスを指定する場合

ジャンプ先のアドレスにシンボルがつけられていて，それがSYMなら，アセンブラ言語は

JP　NZ，SYM

ジャンプ先のアドレスが7DEFなら

JP　NZ，7DEFH

でも大丈夫です。この場合の機械語は

C2・EF・7D

となります。

2．命令の書かれているアドレスとジャンプ先のアドレスとの隔りを指定することによって，ジャンプ先のアドレスを割り出す場合

ジャンプ先のアドレスにシンボルがつけられ，それがSYMなら

JR　NZ，SYM

この命令の書かれている先頭アドレスが7D10で，ジャンプ先のアドレスが7D0Eなら

JR　NZ，7D0EH

でも大丈夫です。この場合の機械語は

0E－10＝（16・14）－（01・00）
＝（15・14）
（15・14）－（00・02）＝（15・12）
＝FC

より

20・FC

となります。詳しくは，無条件ジャンプのところを参照してください。

これらの命令は，各フラグに影響は与えません。

CY＝1ブランチ命令

キャリーフラグの内容を調べ，それが1ならば機械語2ワード目以降で指定するアドレスへジャンプする命令です。

2種類の命令があります。

1．直接ジャンプ先のアドレスを指定する場合

ジャンプ先のアドレスにシンボルがついていて，それがSYMの場合なら，アセンブラ言語は

JP　C，SYM

ジャンプ先のアドレスが7D55なら

JP　C，7D55H

でも大丈夫です。また，このときの機械語は

DA・55・7D

となります。

2．命令の書かれているアドレスと，ジャンプ先のアドレスの隔りを指定することによってジャンプ先のアドレスを割り出す場合

ジャンプ先のアドレスにシンボルがついて，それがSYMなら，アセンブラ言語は

JR　C，SYM

この命令の書かれているアドレスが7D00で，ジャンプ先が7D10の場合なら

JR　C，7D10H

でも大丈夫です。また，このときの機械語は

38・0E

となります。2ワード目の機械語の計算方法は，無条件ジャンプのところを参照してください。

これらの命令は，各フラグには影響を与えません。

CY＝0ブランチ命令

キャリーフラグの内容を調べ，それがゼロなら機械語2ワード目以降で指定するアドレスへジャンプする命令です。

2種類の命令があります。

1，直接ジャンプ先のアドレスを指定する場合

ジャンプ先のアドレスに，シンボルがつけられていて，それがSYMだとすれば，アセンブラ言語は

JP　NC，SYM

ジャンプ先のアドレスが7D27なら

JP　NC，7D27H

でも大丈夫です。また，このときの機械語は，

D2・27・7D

となります。

2．命令の書かれているアドレスと，ジャンプ先のアドレスとの隔りを指定することによって，ジャンプ先のアドレスを割り出す場合

ジャンプ先のアドレスにシンボルがついていてそれがＳＹＭなら

JR　NC，SYM

命令の書かれているアドレスが７D77で，ジャンプ先のアドレスが７D53なら，

JR　NC，７D53H

でも大丈夫です。このときの機械語は

53－77＝(16＋５－１·16＋３)－(07·07)

＝(20·19)－(07·07)

＝(13·12)

さらに２を引いて

(13·12)－(00·02)＝(13·10)

＝DA

より

30·DA

となります。詳しくは，無条件ジャンプのところに説明してありますので，そこを参考にしてください。

これらの命令は，各フラグに影響は与えません。

その他の条件ジャンプ命令

これまでに述べてきた条件ジャンプ命令の他にパリティ奇数ブランチ（同時に，非オーバーフローブランチ），パリティ偶数ブランチ（同時に，オーバーフローブランチ），非負ブランチ，負ブランチ，ＤＪＮＺ命令があります。

ＤＪＮＺ命令は，特殊なものですので，後で独立させて説明することにして，他はまとめて説明することにします。

これらの命令は，それぞれ１通りの命令しかなく，直接ジャンプ先のアドレスを指定しなければなりません。もちろん，アセンブラ言語で書く場合に，シンボルを用いることはできます。

そこで，以下，ジャンプ先のアドレスは，1234とし，そこにはＳＹＭというシンボルがついているものとしましょう。アセンブラ言語の中ではＳＹＭを使いますが，これは1234Hとおきかえても大丈夫です。

・パリティ奇数ブランチ
＝非オーバーフローブランチ

これは，P/Vフラグの内容を調べ，それが，ゼロならば指定されたアドレスへジャンプする命令です。これが，パリティ奇数ブランチなのか非オーバーフローブランチなのかは，その前の命令によります。P/Vフラグが何によって変化したかが問題だからです。

アセンブラ言語，機械語は次のようになります。

JP　PO，SYM

…………E２·34·12

・パリティ偶数ブランチ
＝オーバーフローブランチ

これは，P/Vフラグの内容を調べ，それが１であった場合に指定したアドレスへジャンプする命令です。これが，パリティ偶数ブランチなのか，オーバーフローブランチなのかは，その前の命令によります。

アセンブラ言語，機械語は次のようになります。

JP　PE，SYM

…………EA·34·12

・非負ブランチ

これはＳフラグの内容を調べ，それがゼロならば，指定したアドレスへジャンプする命令です。

アセンブラ言語，機械語は次のようになります。

JP　P，SYM

…………F２·34·12

・負ブランチ

これはＳフラグの内容を調べ，それが１であれば指定したアドレスへジャンプするという命令です。

アセンブラ言語，機械語は次のようになります。

JP　M，SYM

…………FA·34·12

これらの命令は，各フラグには影響を与えません。

ＤＪＮＺ命令

非常に変わった２ワードの命令です。大変便利でもあります。

まず，レジスタＢの内容から１を引きます。そ

して，Bの内容を調べ，Bがもしゼロでなければ，機械語の2ワード目に書かれている数値をプログラムカウンタに加えて，ジャンプし，Bがゼロならば，通常通り，次の命令を実行する，というものです。つまり **第5－1図** に書いたフローチャートを一つの命令で，しかも2ワード命令で，実行してしまう，というものなのです。

7D00番地にこの命令を書き，Bがゼロでないときに，7D12番地へジャンプさせたいのなら

DJNZ　7D12H

…………10・10

となります。7D12番地に，SYMというシンボルがついていれば，アセンブラ言語は

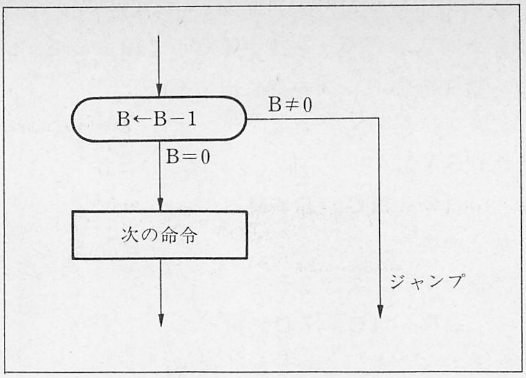
DJNZ　SYM

としてもかまいません。

機械語2ワード目の求め方は，無条件ジャンプのところを参照してください。

各フラグに影響を与えない点にも注意してください。

とにかく大変便利な命令です。



《第5－1図》 DJNZ命令をフローチャートで表わすと

比較命令と条件ジャンプのテストプログラム二つ

(1) パリティ奇数＝非オーバーフローブランチのテストプログラム

おそらく，読者の方々は，「パリィティ奇数＝非オーバーフローブランチ」について，もう少し説明してほしい，と思われていることでしょう。

そこで，この命令を使ったテストプログラムを作って，それを通してもう一度，考えてみることにしましょう。

第5－2図に，そのテストプログラムのフローチャートを書きました。

①の命令を実行すると

A＝11111111　　　（FF）

となって，1になっているビットの数は8で偶数です。したがって，ここでP/V＝1となります。

②で，パリティ奇数ブランチを行ないます。P/V＝1ですから，ジャンプは行なわず，③を実行するはずです。

③で，M(ADR）にAの内容，つまりFFが入ります。

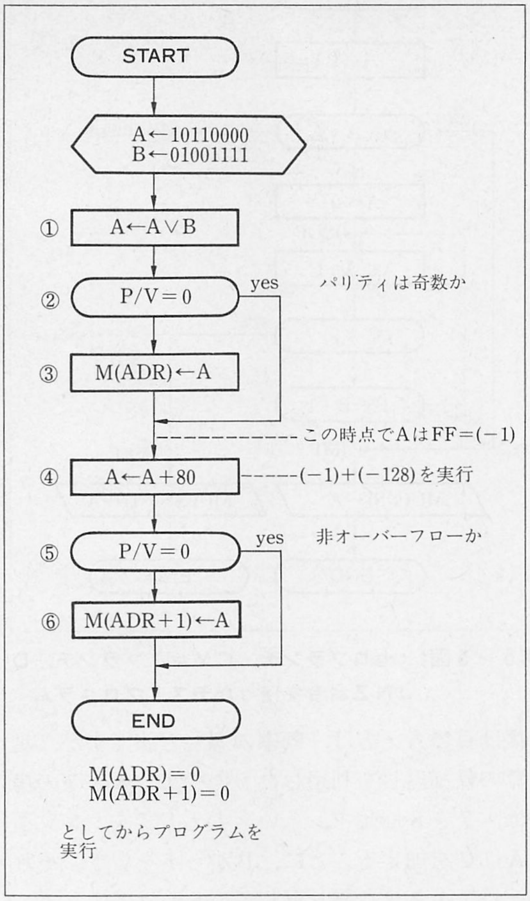
④で，Aに入っているFF，つまり－1と，80つまり－128の和を求めます。

－1＋（－128）＝－129

となって，この数は8ビットで表わすことはできません。したがって，オーバーフローフラグが1になります。…P/V＝1

⑤に，非オーバーフローブランチが，ありま

すが，P/V＝1ですから，ジャンプは行ないません。したがって，⑥を実行します。



《第5－2図》 パリティ奇数＝非オーバーフローブランチをテストするプログラム

⑥Aの値を　M（ADR＋1）に入れます。

FF＋80

の結果，負の範囲でオーバーフローはしていますが，7Fとなっているはずです。

このフローチャートをもとに，アセンブラプログラミングしたものが，**第5－3図**です。

第5－4図は，**第5－3図**より求められた機械語を直接メモリーに書き込み，実行前と実行後のメモリーの内容を比較したものです。

M（ADR）＝FF

M（ADR＋1）＝7F　ただしADR＝7D14

となっていることがわかります。所定通りの働きをしていることが確認できたわけです。

A，Bの値を変えて，パリティが奇数の場合，オーバーフローを起こさない場合について，確かめてみてください。

(2) ゼロブランチ命令，CY＝1ブランチ命令，DJNZ命令を使ったテストプログラム

特に，DJNZ命令の便利な点を知っていただくために，A←B×C　を実行するプログラムを作ってみました。**第5－5図**がそのフローチャートですが，これは，8ビットが0から255の数を表わしているものとしたときの，8ビットどうしの掛け算をするプログラムです。ただし，答も8ビット以内の場合です。8ビットを越える場合は，越えた時点のレジスタAの内容をM(ANS＋1)に送るようになっています。

```
MACRO-80 3.3    09-MAY-79       PAGE    1
                      00100  ;      TEST PROGRAM ( PARITY ODD & NOT OVERFLOW )
                      00200  .Z80
0000'                 00300  ASEG
                      00400  ORG    7D00H
                      00500  ;
7D00    3E B0         00800         LD      A,10110000B     ; A=10110000
7D02    06 4F         00900         LD      B,01001111B     ; B=01001111
7D04    B0            01000         OR      B               ; A=( A OR B )
7D05    E2 7D0B       01100         JP      PO,SYM1         ; IF P/V=0 GOTO SYM1
7D08    32 7D14       01200         LD      (ADR),A         ; M(ADR)=A
7D0B    C6 80         01300  SYM1:  ADD     A,80H           ; A=A+80
7D0D    E2 7D13       01400         JP      PO,SYM2         ; IF P/V=0 GOTO SYM2
7D10    32 7D15       01500         LD      (ADR+1),A       ; M(ADR+1)=A
7D13    C9            01600  SYM2:  RET                     ; RETURN
7D14    00            01700  ADR:   DB      00H             ; M(ADR)=0
7D15    00            01800         DB      00H             ; M(ADR+1)=0
                      01900         END
```

《第5－3図》 第5－2図のフローチャートをもとにアセンブラプログラミングしたもの

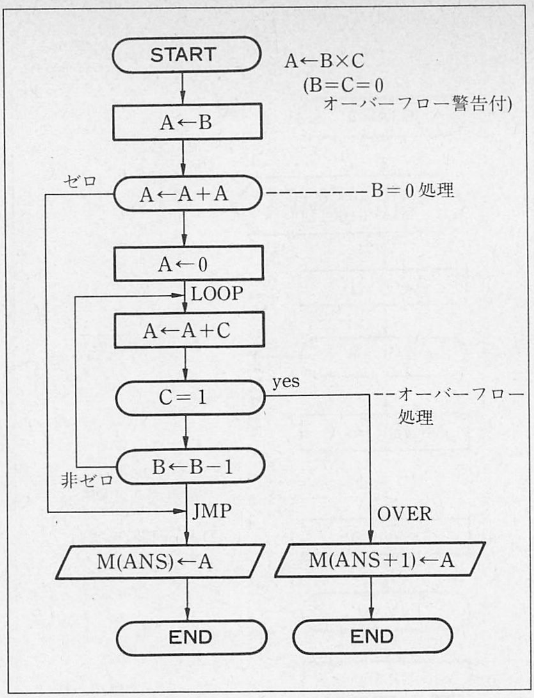
** BEFORE RUN

NO 1	7D00	3E
NO 2	7D01	08
NO 3	7D02	06
NO 4	7D03	4F
NO 5	7D04	B0
NO 6	7D05	E2
NO 7	7D06	0B
NO 8	7D07	7D
NO 9	7D08	32
NO 10	7D09	14
NO 11	7D0A	7D
NO 12	7D0B	C6
NO 13	7D0C	80
NO 14	7D0D	E2
NO 15	7D0E	13
NO 16	7D0F	7D
NO 17	7D10	32
NO 18	7D11	15
NO 19	7D12	7D
NO 20	7D13	C9
NO 21	7D14	00
NO 22	7D15	00

** AFFTER RUN

NO 1	7D00	3E
NO 2	7D01	08
NO 3	7D02	06
NO 4	7D03	4F
NO 5	7D04	B0
NO 6	7D05	E2
NO 7	7D06	0B
NO 8	7D07	7D
NO 9	7D08	32
NO 10	7D09	14
NO 11	7D0A	7D
NO 12	7D0B	C6
NO 13	7D0C	80
NO 14	7D0D	E2
NO 15	7D0E	13
NO 16	7D0F	7D
NO 17	7D10	32
NO 18	7D11	15
NO 19	7D12	7D
NO 20	7D13	C9
NO 21	7D14	FF
NO 22	7D15	7F

《第５－４図》 第５－３図の機械語を直接メモリーに書き込み，実行前と実行後のメモリーの内容を比較したもの



《第５－５図》 ゼロブランチ，ＣＹ＝１ブランチ，ＤＪＮＺ命令を使ったテストプログラム

掛け算の考え方は，効率の悪い方法ですが，足し算の繰り返しを利用したものです。８×７の場合は，７を８回足す，という考え方です。レジスタＡにＣを加えるごとに，Ｂから１を引き，それがゼロになるまで繰り返しているわけです。ここにＤＪＮＺ命令が効率よく使われています。

第５－６図は，８×７を実行する場合のプログラムを，アセンブラで作ってみたものです。

第５－７図は**第５－６図**の機械語を直接メモリーに書き込み，実行する前後のメモリーの内容を比較したものです。８×７の結果，７Ｄ16番地が38，つまり10進数の56になっていることがわかります。

第５－８図は，**第５－７図**の実行後，７Ｄ０１の内容をゼロにして，つまり Ｂ＝０ にして，０×７を実験するようにし，また実行したものです。７Ｄ16番地がゼロに変わりました。

第５－９図は**第５－８図**の実行後，７Ｄ０１の内容を25，（10進数で37）に変え，37×７を実行しようとし，また実行したものです。

37×７＝259 ですから，255 を上まわり，256(＝０）を３上まわります。したがって，７Ｄ17番地に０３が入っているわけです。

```
MACRO-80 3.3    09-MAY-79         PAGE    1
                              00100  ;       TEST PROGRAM ( DJNZ )
                              00200  .Z80
0000'                         00300  ASEG
                              00400  ORG     7D00H
                              00500  ;
7D00    06 08                 00600          LD      B,8H          ; B=8
7D02    0E 07                 00700          LD      C,7H          ; C=7
                              00800  ;
7D04    78                    00900          LD      A,B           ; A=B
7D05    B7                    00910          OR      A             ; A=( A OR A )
7D06    28 06                 00920          JR      Z,JMP         ; IF A=0 GOTO JMP
7D08    97                    00930          SUB     A             ; A=0
7D09    81                    01000  LOOP:   ADD     A,C           ; A=A+C
7D0A    38 06                 01050          JR      C,OVER        ; IF CY=1 GOTO OVER
7D0C    10 FB                 01100          DJNZ    LOOP          ; B=B-1 : IF B<>0 GOTO LOOP
7D0E    32 7D16               01200  JMP:    LD      (ANS),A       ; M(ANS)=A
7D11    C9                    01300          RET                   ; RETURN
                              01400  ;
7D12    32 7D17               01500  OVER:   LD      (ANS+1),A     ; M(ANS+1)=A
7D15    C9                    01600          RET                   ; RETURN
                              01700  ;
7D16    00                    01800  ANS:    DB      0H            ; M(ANS)=0
7D17    00                    01900          DB      0H            ; M(ANS+1)=0
                              02000          END
```

《第５－６図》 第５－５図をもとに，アセンブラプログラミングしたもの

** BEFORE RUN

NO 1	7D00	06
NO 2	7D01	08
NO 3	7D02	0E
NO 4	7D03	07
NO 5	7D04	78
NO 6	7D05	B7
NO 7	7D06	28
NO 8	7D07	06
NO 9	7D08	97
NO 10	7D09	81
NO 11	7D0A	38
NO 12	7D0B	06
NO 13	7D0C	10
NO 14	7D0D	FB
NO 15	7D0E	32
NO 16	7D0F	16
NO 17	7D10	7D
NO 18	7D11	C9
NO 19	7D12	32
NO 20	7D13	17
NO 21	7D14	7D
NO 22	7D15	C9
NO 23	7D16	00
NO 24	7D17	00

** AFFTER RUN

NO 1	7D00	06
NO 2	7D01	08
NO 3	7D02	0E
NO 4	7D03	07
NO 5	7D04	78
NO 6	7D05	B7
NO 7	7D06	28
NO 8	7D07	06
NO 9	7D08	97
NO 10	7D09	81
NO 11	7D0A	38
NO 12	7D0B	06
NO 13	7D0C	10
NO 14	7D0D	FB
NO 15	7D0E	32
NO 16	7D0F	16
NO 17	7D10	7D
NO 18	7D11	C9
NO 19	7D12	32
NO 20	7D13	17
NO 21	7D14	7D
NO 22	7D15	C9
NO 23	7D16	38
NO 24	7D17	00

《第５－７図》 第５－６図の機械語を直接メモリーに入れ，８×７を実験する前と後のメモリーの内容

```
** BEFORE RUN
----------------------------------------
NO 1      7D00      06
NO 2      7D01      00
NO 3      7D02      0E
NO 4      7D03      07
NO 5      7D04      78
NO 6      7D05      B7
NO 7      7D06      28
NO 8      7D07      06
NO 9      7D08      97
NO 10     7D09      81
NO 11     7D0A      38
NO 12     7D0B      06
NO 13     7D0C      10
NO 14     7D0D      FB
NO 15     7D0E      32
NO 16     7D0F      16
NO 17     7D10      7D
NO 18     7D11      C9
NO 19     7D12      32
NO 20     7D13      17
NO 21     7D14      7D
NO 22     7D15      C9
NO 23     7D16      38
NO 24     7D17      00
----------------------------------------
** AFFTER RUN
----------------------------------------
NO 1      7D00      06
NO 2      7D01      00
NO 3      7D02      0E
NO 4      7D03      07
NO 5      7D04      78
NO 6      7D05      B7
NO 7      7D06      28
NO 8      7D07      06
NO 9      7D08      97
NO 10     7D09      81
NO 11     7D0A      38
NO 12     7D0B      06
NO 13     7D0C      10
NO 14     7D0D      FB
NO 15     7D0E      32
NO 16     7D0F      16
NO 17     7D10      7D
NO 18     7D11      C9
NO 19     7D12      32
NO 20     7D13      17
NO 21     7D14      7D
NO 22     7D15      C9
NO 23     7D16      00
NO 24     7D17      00
----------------------------------------
```

《第５－８図》 第５－７図実行後，7D01 を００とし，０×７を実験する前後のメモリーの内容

```
** BEFORE RUN
----------------------------------------
NO 1      7D00      06
NO 2      7D01      25
NO 3      7D02      0E
NO 4      7D03      07
NO 5      7D04      78
NO 6      7D05      B7
NO 7      7D06      28
NO 8      7D07      06
NO 9      7D08      97
NO 10     7D09      81
NO 11     7D0A      38
NO 12     7D0B      06
NO 13     7D0C      10
NO 14     7D0D      FB
NO 15     7D0E      32
NO 16     7D0F      16
NO 17     7D10      7D
NO 18     7D11      C9
NO 19     7D12      32
NO 20     7D13      17
NO 21     7D14      7D
NO 22     7D15      C9
NO 23     7D16      00
NO 24     7D17      00
----------------------------------------
** AFFTER RUN
----------------------------------------
NO 1      7D00      06
NO 2      7D01      25
NO 3      7D02      0E
NO 4      7D03      07
NO 5      7D04      78
NO 6      7D05      B7
NO 7      7D06      28
NO 8      7D07      06
NO 9      7D08      97
NO 10     7D09      81
NO 11     7D0A      38
NO 12     7D0B      06
NO 13     7D0C      10
NO 14     7D0D      FB
NO 15     7D0E      32
NO 16     7D0F      16
NO 17     7D10      7D
NO 18     7D11      C9
NO 19     7D12      32
NO 20     7D13      17
NO 21     7D14      7D
NO 22     7D15      C9
NO 23     7D16      00
NO 24     7D17      03
----------------------------------------
```

《第５－９図》 第５－８図の実行後，7D01 を25にして，37×７を実験する前後のメモリーの内容 （オーバーフローの場合）

6 ローテイト命令とシフト命令

ここでは，レジスタやメモリーにある8個のビットのもつ情報を回転させたり，シフトさせたりする命令をあつかいます。

RLC命令（RLCAを含む）

RLCのRは，ROTATEのRです。Lは，LEFTのL，Cは，CIRCULARのCです。

これでおわかりのように，8個のビットを一つの輪のように考えて，左に回転させる命令です。

この命令の前に，あるレジスタまたはメモリーの8ビットが

10110001 ……（B1）

であったとすると，この命令を実行した後には

01100011 ……（63）

となります。それぞれのビットの持つ情報が左に移され，最上位ビットの情報が最下位ビットに送られているわけです。

この種の命令は，対象になるレジスタ，メモリーの違いで，11種類あります。ただし，レジスタAに関しては，ほぼ同じ命令が二つあります。一方は1ワード命令，一方は2ワード命令ですが，フラグに与える影響が違います。これは，Z80が，8080上位コンパチブルに作られていることによるものです。

アセンブラ言語，機械語の順にまとめると次のようになります。ただし，IX，IYに加える数は，16進数の23としてあります。

RLC B …………………… CB・00
RLC C …………………… CB・01
RLC D …………………… CB・02
RLC E …………………… CB・03
RLC H …………………… CB・04
RLC L …………………… CB・05
RLC A …………………… CB・07
RLCA ……………………………… 07
（RLCAと，RLC，Aとは，ほぼ同じ働きをする。）
RLC （HL）……………… CB・06
RLC （IX＋23H）
……… DD・CB・23・06
RLC （IY＋23H）
……… FF・CB・23・06

上で，おわかりのように，インデックスレジス

タを用いるときの機械語は，４ワードにまたがります。そして，インデックスレジスタに加える数，この場合は２３ですが，それが第３ワード目にきます。

これらの命令が，各フラグに与える影響は次の通りです。

S ： ＲＬＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの８個のビットが１ならば，Ｓ＝１，そうでなければ，Ｓ＝０

Z ： ＲＬＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの全ビットがゼロならば，Ｚ＝１，そうでなければＺ＝ϕ 。

P/V： ＲＬＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの８個のビットのうち，１になっているものが偶数個ならば，Ｐ/Ｖ＝１，そうでなければ，Ｐ/Ｖ＝０

CY： 実行前に，指定したレジスタあるいはメモリーの最上位ビットが１ならば，ＣＹ＝１，そうでなければ，ＣＹ＝０

ＲＲＣ命令
（ＲＲＣＡを含む）

ＲＲＣのはじめのＲは，ＲＯＴＡＴＥのＲ，次のＲは，ＲＩＧＨＴのＲ，Ｃは，ＣＩＲＣＵＬＡＲのＣです。これからおわかりのように，この命令は，レジスタなりメモリーなりの８個のビットを一つの輪のように考えて，それを右まわりに回転させる命令です。

たとえば，

　　１０１１００１０

に対して，この命令を実行すると，

　　０１０１１００１

となります。最下位ビットの情報が最上位ビットに送られ，他は一つずつ右どなりに送られるわけです。

全部で11種類の命令があります。アセンブラ言語，機械語の順でまとめると，次のようになります。ただし，ＩＸ，ＩＹに加える数を16進数ＡＢとしてあります。また，ＲＲＣＡとＲＲＣ，Ａとは，フラグに与える影響が違うことと，前者が１ワード命令なのに対して，後者は２ワード命令であることが違いますが，他の働きは同じです。

ＲＲＣ　Ｂ　…………………　ＣＢ・０８
ＲＲＣ　Ｃ　…………………　ＣＢ・０９
ＲＲＣ　Ｄ　…………………　ＣＢ・０Ａ
ＲＲＣ　Ｅ　…………………　ＣＢ・０Ｂ
ＲＲＣ　Ｈ　…………………　ＣＢ・０Ｃ
ＲＲＣ　Ｌ　…………………　ＣＢ・０Ｄ
ＲＲＣ　Ａ　…………………　ＣＢ・０Ｆ
ＲＲＣＡ　……………………………　０Ｆ
ＲＲＣ　（ＨＬ）　……………　ＣＢ・０Ｅ
ＲＲＣ　（ＩＸ＋０ＡＢＨ）
　　　………　ＤＤ・ＣＢ・ＡＢ・０Ｅ
ＲＲＣ　（ＩＸ＋０ＡＢＨ）
　　　………　ＦＤ・ＣＢ・ＡＢ・０Ｅ

これらの命令が，各フラグに与える影響は次の通りです。

S ： ＲＲＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの最上位のビットが１ならば，Ｓ＝１，そうでなければＳ＝０

Z ： ＲＲＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの最上位ビットすべてがゼロならば，Ｚ＝１，そうでなければ，Ｚ＝０

P/V： ＲＲＣＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの８個のビットのうち，１になっているものの数が偶数ならば，Ｐ/Ｖ＝１，そうでなければＰ/Ｖ＝０

CY： 実行前，指定したレジスタあるいはメモリーの最下位ビット（第ゼロビット）が１ならば，ＣＹ＝１，そうでなければ，ＣＹ＝０

ＲＬ命令
（ＲＬＡを含む）

ＲＬのＲは，ＲＯＴＡＴＥ，Ｌは，ＬＥＦＴの意味です。これからおわかりのように，レジスタを左に回転させるのですが，実はキャリーを含めて，一つの輪を作りそれを左に回転させる命令なのです。たとえば

ＣＹ＝１，　０１１０００１１

に対して，このＲＬ命令を実行すると，

ＣＹ＝０　　００１００１１１

となるのです。キャリーの持つ情報が，指定したレジスタなりメモリーの最下位ビットに入り，最上位のビットの情報がキャリーに送られ，あとは一つずつ左に送られるのです。

この命令は全部で11種類あり，次のようになります。ただし，ＲＬＡと，ＲＬ，Ａ　はフラグに与える影響が違うのと，前者が１ワード命令であることを除けば同じ命令です。またＩＸ，ＩＹに加える値は，ここでは，16進数　56　としてあります。

ＲＬ　Ｂ	ＣＢ・１０
ＲＬ　Ｃ	ＣＢ・１１
ＲＬ　Ｄ	ＣＢ・１２
ＲＬ　Ｅ	ＣＢ・１３
ＲＬ　Ｈ	ＣＢ・１４
ＲＬ　Ｌ	ＣＢ・１５
ＲＬ　Ａ	ＣＢ・１７
ＲＬＡ	１７
ＲＬ　（ＨＬ）	ＣＢ・１６
ＲＬ　（ＩＸ＋５６Ｈ）	ＤＤ・ＣＢ・５６・１６
ＲＬ　（ＩＹ＋５６Ｈ）	ＦＤ・ＣＢ・５６・１６

これらの命令が各フラグに与える影響は次の通りです。

Ｓ：　ＲＬＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの最上位ビットが１ならば，Ｓ＝１，そうでなければ，Ｓ＝０

Ｚ：　ＲＬＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの８個のビットすべてがゼロならば，Ｚ＝１，そうでなければ，Ｚ＝０

Ｐ/Ｖ：　ＲＬＡの場合は，影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの８個のビットのうち１になっているものの数が偶数ならば，Ｐ/Ｖ＝１，そうでなければ，Ｐ/Ｖ＝０

ＣＹ：　実行前，指定したレジスタなりメモリーの最上位のビットが１ならば，ＣＹ＝１，そうでなければ，ＣＹ＝０

ＲＲ命令
（ＲＲＡを含む）

ＲＲは右に回転の意味ですが，実はキャリーを含めての回転です。たとえば，

ＣＹ＝１，　００１１１１１０

に対して，この命令を行なうと，

ＣＹ＝０　　１００１１１１１

となります。

キャリーの情報が，指定したレジスタなり，メモリーの最上位ビットへ送られ，以下一つずつ右に送られ，そして，最下位ビットの情報がキャリーに送られるのです。

これらの命令は全部で11種類あります。ただしＲＲＡとＲＲ　Ａは，フラグへ与える影響が違い，前者が１ワード命令であることを除けば同じ命令です。

以下，これらの命令をまとめてみますが，インデックスレジスタに加える値は，ここでは16進数の９８としておきます。

ＲＲ　Ｂ	ＣＢ・１８
ＲＲ　Ｃ	ＣＢ・１９
ＲＲ　Ｄ	ＣＢ・１Ａ
ＲＲ　Ｅ	ＣＢ・１Ｂ
ＲＲ　Ｈ	ＣＢ・１Ｃ
ＲＲ　Ｌ	ＣＢ・１Ｄ
ＲＲ　Ａ	ＣＢ・１Ｆ
ＲＲＡ	１Ｆ
ＲＲ　（ＨＬ）	ＣＢ・１Ｅ

RR　(IX+98H)
…………　DD・CB・98・1E
RR　(IY+98H)
…………　FD・CB・98・1E

これらの命令が，各フラグに与える影響は次の通りです。

S　：　RRAの場合は影響をうけない。他の場合は，実行後，指定した，レジスタあるいはメモリーの最上位のビットが1ならば，S＝1，そうでなければ，S＝0

Z　：　RRAの場合は影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの8個のビットすべてがゼロならば，Z＝1，そうでなければ，Z＝0

P/V：　RRAの場合は影響をうけない。他の場合は，実行後，指定したレジスタあるいはメモリーの8個のビットのうち1になっているものが偶数個ならば，P/V＝1，そうでなければ，P/V＝0

CY：　実行前，指定したレジスタあるいはメモリの最下位ビット（第ゼロビット）が1ならば，CY＝1，そうでなければ，CY＝0

SLA命令
（2倍命令）

SLAのSは，SHIFTのS，Lは，LEFTのL，Aは，ARITHMETICのAです。

なぜ，そんな名前がついたのか，といいますと，このシフト命令を実行すると，丁度，もとの数値を2倍することと同じになるからです。たとえば，

00000001　（つまり1）

に対してこの命令を実行すると

00000010　（つまり2）

さらに，この命令を実行すると

00000100　（つまり4）

というふうになるわけです。

具体的には，第7ビットの情報をキャリーにいれ，第6ビットから第1ビットまでのそれを一つ左のビットに移し，第1ビットにはゼロを入れる，ということをします。これで，正負を考慮した上で，2倍しても，オーバーフローを起こさない範囲の，つまり，−64以上，＋63以下の範囲のすべての数は，2倍されることになります。

この命令は，対象とするレジスタ，メモリーの違いで，10種類あります。

以下，それを記号，アセンブラ言語，機械語の順でまとめますが，インデックスレジスタIX，IYに加える数は，ここでは16進数　72　としておきます。

B←2＊B
　SLA　B　………………　CB・20
C←2＊C
　SLA　C　………………　CB・21
D←2＊D
　SLA　D　………………　CB・22
E←2＊E
　SLA　E　………………　CB・23
H←2＊H
　SLA　H　………………　CB・24
L←2＊L
　SLA　L　………………　CB・25
A←2＊A
　SLA　A　………………　CB・27
M(HL)←2＊M(HL)
　SLA　(HL)　………　CB・26
M(IX+72)←2＊M(IX+72)
　SLA　(IX+72H)
………………　DD・CB・72・26
M(IY+72)←2＊M(IY+72)
　SLA　(IY+72H)
………　FD・CB・72・26

これらの命令が各フラグに与える影響は，次の通りです。

S　：　実行後，指定したアドレスまたはメモリーの第7ビットが1ならば，S＝1，そうでなければ，S＝0

Z　：　実行後，指定したアドレスまたはメモリーの8個あるビットすべてがゼロならば，Z＝1，そうでなければ，Z＝0

P/V：　特に注意が必要です。ARITHMETICのAがついた命令ですが，ここでは，

オーバーフローの有無ではなく，パリティによって，R/Vが変化します。つまり，実行後，指定したレジスタやメモリーの8個あるビットのうち，1になっているものの数が偶数ならば，P/V＝1，そうでなければ，P/V＝0

CY： 実行前に，指定したレジスタあるいはメモリーの第7ビットが1ならば，CY＝1，そうでなければ，CY＝0

SRA命令
（1/2命令 その1）

SRAのSは，SHIFTのS，Rは，RIGHTのR，そしてAはARITHMETICのAです。

具体的には，レジスタなりメモリーの最下位のビットの持つ情報をキャリーに送り，第2から第7ビットまでのそれを一つずつ右に移し，そして，第7ビットには何を入れるかといいますと，命令実行前の第7ビットに入っていたものを，ここにも入れるのです。したがって，実行後は，必ず，第7ビットと第6ビットは同じ値になります。たとえば，

10110010 …… （－78）

に対して，この命令を実行すると

11011001 …… （－39）

となります。

すでにおわかりのように，前の内容の第1ビットが1でなければ，命令実行後は，正の数も負の数も丁度半分になります。

ただし，8ビットの表わす数が，すべて正の数と考える場合には，うまくいかないので注意してください。そういった場合には次にとりあげるSRL命令が用意されています。

さて，この命令は全部で10種類あります。以下にまとめてみますが，IX，IYを用いる場合については，それに加える数を16進数A3としてあります。

B←B／2（AR）
　SRA B ……………… CB・28
C←C／2（AR）
　SRA C ……………… CB・29
D←D／2（AR）
　SRA D ……………… CB・2A
E←E／2（AR）
　SRA E ……………… CB・2B
H←H／2（AR）
　SRA H ……………… CB・2C
L←L／2（AR）
　SRA L ……………… CB・2D
A←A／2（AR）
　SRA A ……………… CB・2F
M（HL）←M（HL）／2（AR）
　SRA （HL） ……… CB・2E
M（IX＋A3）←M（IX＋A3）／2（AR）
　SRA （IX＋0A3H）
　…………… DD・CB・A3・2E
M（IY＋A3）←M（IY＋A3）/2（AR）
　SRA （IY＋0A3H）
　…………… FD・CB・A3・2E

これらの命令が各フラグに与える影響は，次の通りです。

S ： 実行前に，指定されたレジスタあるいはメモリーの第7ビットが1ならば，S＝1，そうでなければ，S＝0

Z ： 実行後，指定されたレジスタあるいはメモリーの8個のビットがすべてゼロの場合，Z＝1，そうでなければ，Z＝0

P/V: 注意してください。ARITHMETICのAがついた命令ですが，P/Vはパリティで変化します。つまり，指定されたレジスタあるいはメモリーの8個あるビットのうち，実行後に1であるものの数が偶数ならば，P/V＝1，そうでなければ，P/V＝0

CY： 実行前，指定したレジスタあるいはメモリーの第ゼロビット（最下位ビット）が1ならば，CY＝1，そうでなければ，CY＝0

SRL命令
（1/2命令 その2）

8ビットの数が，－128から＋127までの数を表わしているとしたときの1/2命令は前にお話しし

ました。ここでは，８ビットの数が０から＋255までの数を表わしているとしたときの1/2命令をお話しします。

ＳＲＬ命令といいますが，ＳはSHIFTのS, RはRIGHTのR, Lは, LOGICALのLです。

具体的には，第ゼロビットの持つ情報をキャリーに送り，第１ビットから第７ビットの情報はそれぞれ一つ右のビットに入れ，そして，第７ビットにはゼロを入れるという働きをします。

レジスタなりメモリーに入っている数が偶数ならば，この命令を実行すると，もとの1/2になります。奇数の場合は，２で割って切り捨となります。

たとえば，

　　１１０１１０１０　　　（＋218）

に対してこの命令を実行すると，

　　０１１０１１０１　　　（＋109）

となります。

もとの値が奇数であった場合は，必ずキャリーが１になるので，後から確認をしたり四捨五入をすることも容易です。

この種の命令は全部で10種類あり，まとめると次のようになります。ただし，ＩＸ，ＩＹに加える数は，４としてあります。

Ｂ←Ｂ／２（ＬＧ）
　ＳＲＬ　Ｂ　………………　ＣＢ・３８

Ｃ←Ｃ／２（ＬＧ）
　ＳＲＬ　Ｃ　………………　ＣＢ・３９

Ｄ←Ｄ／２（ＬＧ）
　ＳＲＬ　Ｄ　………………　ＣＢ・３Ａ

Ｅ←Ｅ／２（ＬＧ）
　ＳＲＬ　Ｅ　………………　ＣＢ・３Ｂ

Ｈ←Ｈ／２（ＬＧ）
　ＳＲＬ　Ｈ　………………　ＣＢ・２Ｃ

Ｌ←Ｌ２（ＬＧ）
　ＳＲＬ　Ｌ　………………　ＣＢ・２Ｄ

Ａ←Ａ／２（ＬＧ）
　ＳＲＬ　Ａ　………………　３Ｂ・２Ｆ

Ｍ(ＨＬ）←Ｍ(ＨＬ）／２（ＬＧ）
　ＳＲＬ　（ＨＬ）　………　３Ｂ・２Ｅ

Ｍ(ＩＸ＋４）←Ｍ（ＩＸ＋４）／２（ＬＧ）
　ＳＲＬ　（ＩＸ＋４Ｈ）
　　………　ＤＤ・ＣＢ・０４・２Ｅ

Ｍ(ＩＹ＋４）←Ｍ（ＩＹ＋４）／２（ＬＧ）
　ＳＲＬ　（ＩＹ＋４Ｈ）
　　………　ＦＤ・ＣＢ・０４・２Ｅ

これらの命令が，各フラグに与える影響は次の通りです。

Ｓ　：　必ずゼロ

Ｚ　：　実行後，指定されたレジスタあるいはメモリーの８個のビットすべてがゼロの場合, Ｚ＝１, そうでなければＺ＝０

P/V：　実行後，指定されたレジスタあるいはメモリーの８個のビットのうち，１になっているものの数が偶数個であればＰ／Ｖ＝１, そうでなければ, Ｐ／Ｖ＝０

ＣＹ：　実行前に，指定されたレジスタあるいはメモリーの最下位ビットが１ならば, ＣＹ＝１, そうでなければ, ＣＹ＝０

ＲＬＤ命令

ＲＬＤのＲは，ＲＯＴＡＴＥのＲ，ＬはＬＥＦＴのＬ，そして，ＤはＤＩＧＩＴのＤです。

４ビット単位で左に回転させる命令です。１ビット単位のときにキャリーが重要な役をしていましたが，この命令の場合，それに代るものとして，レジスタＡの下４ビットが使われています。

対象となる８ビットは，ＨＬレジスタペアによってアドレスを指定されるメモリーのものです。

具体的に考えてみましょう。命令実行前各レジスタ，メモリーが次のようであったとします。

　Ａ＝２３
　ＨＬ＝７Ｄ００
　Ｍ（７Ｄ００）＝４５

数値はいずれも16進数です。この命令を実行すると，Ｍ（７Ｄ００）の上位４ビットで表わされている４が，レジスタＡの下位４ビットに入り，レジスタＡの下位４ビットで表わされていた３が，Ｍ(７Ｄ００)の下位４ビットに入り, Ｍ(７Ｄ００）の下位４ビットで表わされていた５が，そのメモリーの上位４ビットに入ります。ＨＬの内容は変わりません。したがって，次のようになります。

　Ａ＝２４
　ＨＬ＝７Ｄ００
　Ｍ（７Ｄ００）＝５３

この命令は，連続するメモリーに，何桁かの10進数（BCDコードの）が入っているときに，10進数の桁単位でのシフトをするときに大変便利なのです。

たとえば，

M（7D00）＝45
M（7D01）＝23
M（7D02）＝01

を

M（7D00）＝50
M（7D01）＝34
M（7D02）＝12

とする場合には，次のようにすればよいわけです。

① A←00
②HL←7D00
③RLD命令
④HL←HL＋1
⑤RLD命令
⑥HL←HL＋1
⑦RLD命令

③において，A＝04，M（7D02）＝50
⑤において　A＝02，M（7D01）＝34
⑦において　A＝00，M（7D00）＝12

となることは，もうおわかりでしょう。

この命令のアセンブラ言語は，

RLD

だけで，機械語は，

ED・6F

です。

この命令が各シフトに与える影響も，ちょっと変則ですから注意してください。それは以下のようになります。

S　：　実行後，レジスタAの最上位ビットが1ならば，S＝1，そうでなければS＝0
Z　：　実行後，レジスタAの8個のビットがすべてゼロなら，Z＝1，そうでなければ，Z＝0
P/V：　実行後，レジスタAの8個のビットのうち，1になっているものが偶数個あれば，P/V＝1，そうでなければ，P/V＝0
CY：　影響はうけない

RRD命令

RLD命令と回転方向が逆の命令です。詳しくは，RLD命令を参照してください。

命令実行前に

A＝27
HL＝8204
M（8204）＝04

であれば，実行後は，

A＝24
HL＝8204
M（8204）＝70

となります。つまり，レジスタAの下位4ビットで表わされている7が，M（8204）の上位4ビットに，M（8204）の上位4ビットで表わされていた0が，その下位4ビットに，そして，その下位4ビットで表わされていた4が，レジスタAの下位4ビットに，それぞれ移るという命令です。

この命令をアセンブラ言語で表わすと，

RRD

となり，機械語は

ED・67

です。

この命令が，各フラグに与える影響は，RLD命令と全く同じですから，そこを参照してください。

ローテイト・シフト命令を使ったプログラム二つ

ローテイト命令やシフト命令の働きを，具体的に見れるようなプログラムを考えてみましょう。まずは，ＲＬＣＡ命令と，ＲＬＣ Ａ 命令の違いをみるためのプログラムです。

第６－１図にフローチャートを書いてみました。ＲＬＣＡと ＲＬＣ Ａの違いは，本文中にも説明しましたが，ＲＬＣＡの方は，キャリー以外のフラグに影響を与えません。そこでレジスタＡに７０，２進数なら，０１１１００００ですが，それを入れて，ＲＬＣＡとＲＬＣ Ａの二つの命令を実行してみます。実行後，レジスタＡの内容は，２進数で，１１１０００００になるはずです。

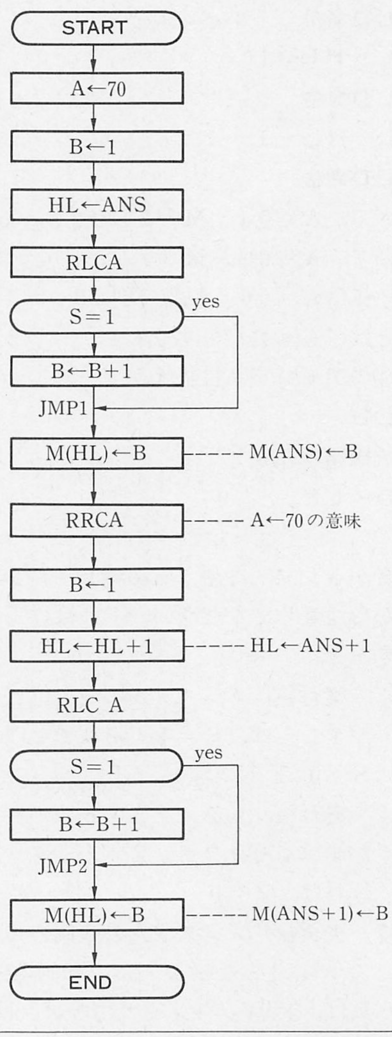
ここで，このレジスタの第７ビットは0から１に変化しました。ですから，もしこれらの命令が各種フラグに影響を与えるとすれば，Ｓフラグは１になるはずです。

そこで，Ｓフラグをチェックし，もしそれが１ならば，レジスタＢに1が，もし，それがゼロのままなら，レジスタＢに２が入るようなプログラムを作れば，実際に，ＲＬＣＡとＲＬＣ Ａの違いを見ることができます。

フローチャートからおわかりだと思いますが，ＡＮＳのシンボルの付けられたメモリーには，２が，その次のメモリーには，１が入るはずです。

第６－２図は，**第６－１図**のフローチャートをもとに，アセンブラプログラムをしてみたものです。プログラムは ７Ｄ００ から書き始める関係で，ＡＮＳは，７Ｄ19を指すことになりました。

何度もお話ししましたが，アドレス ７Ｄ０４，



《第６－１図》 ＲＬＣＡとＲＬＣ Ａのちがいを調べるプログラム

```
MACRO-80 3.3    09-MAY-79        PAGE    1
                                 00100  ;        TEST PROGRAM ( RLCA & RLC A )
                                 00200  .Z80
0000'                            00300  ASEG
                                 00400  ORG      7D00H
                                 00500  ;
7D00    3E 70                    00600           LD      A,70H              ; A=70
7D02    06 01                    00700           LD      B,1H               ; B=1
7D04    21 7D19                  00750           LD      HL,ANS'            ; HL=ANS
7D07    07                       00800           RLCA
7D08    FA 7D0C                  00900           JP      M,JMP1             ; IF S=1 GOTO JMP1
7D0B    04                       01000           INC     B                  ; B=B+1
7D0C    70                       01100  JMP1:    LD      (HL),B             ; M(ANS)=B
                                 01200  ;
7D0D    0F                       01300           RRCA                       ; A=70
7D0E    06 01                    01400           LD      B,1H               ; B=1
7D10    23                       01450           INC     HL                 ; HL=HL+1
7D11    CB 07                    01500           RLC     A
7D13    FA 7D17                  01600           JP      M,JMP2             ; IF S=1 GOTO JMP2
7D16    04                       01700           INC     B                  ; B=B+1
7D17    70                       01800  JMP2:    LD      (HL),B             ; M(ANS+1)=B
7D18    C9                       01900           RET                        ; RETURN
                                 02000  ;
7D19    00                       02100  ANS:     DB      0H                 ; M(ANS1)=0
7D1A    00                       02200           DB      0H                 ; M(ANS+1)=0
                                 02300           END
```

《第6－2図》 第6－1図のフローチャートをもとに，アセンブラプログラムをしたもの

** BEFORE RUN

NO 1	7D00	3E
NO 2	7D01	70
NO 3	7D02	06
NO 4	7D03	01
NO 5	7D04	21
NO 6	7D05	19
NO 7	7D06	7D
NO 8	7D07	07
NO 9	7D08	FA
NO 10	7D09	0C
NO 11	7D0A	7D
NO 12	7D0B	04
NO 13	7D0C	70
NO 14	7D0D	0F
NO 15	7D0E	06
NO 16	7D0F	01
NO 17	7D10	23
NO 18	7D11	CB
NO 19	7D12	07
NO 20	7D13	FA
NO 21	7D14	17
NO 22	7D15	7D
NO 23	7D16	04
NO 24	7D17	70
NO 25	7D18	C9
NO 26	7D19	00
NO 27	7D1A	00

** AFFTER RUN

NO 1	7D00	3E
NO 2	7D01	70
NO 3	7D02	06
NO 4	7D03	01
NO 5	7D04	21
NO 6	7D05	19
NO 7	7D06	7D
NO 8	7D07	07
NO 9	7D08	FA
NO 10	7D09	0C
NO 11	7D0A	7D
NO 12	7D0B	04
NO 13	7D0C	70
NO 14	7D0D	0F
NO 15	7D0E	06
NO 16	7D0F	01
NO 17	7D10	23
NO 18	7D11	CB
NO 19	7D12	07
NO 20	7D13	FA
NO 21	7D14	17
NO 22	7D15	7D
NO 23	7D16	04
NO 24	7D17	70
NO 25	7D18	C9
NO 26	7D19	02
NO 27	7D1A	01

《第6－3図》 第6－2図の機械語を直接メモリーに入れ，実行前と実行後のメモリーの内容を比較したもの

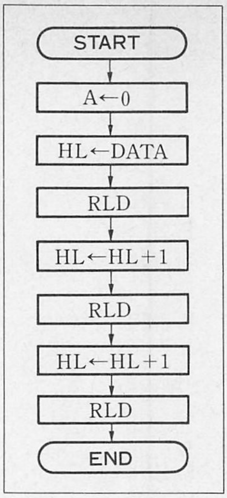
7D08，7D13から始まる機械語は，実際にメモリーに入れる場合，次の順で入れなければなりません。

7D04…21·19·7D
7D08…FA·0C·7D
7D13…FA·17·7D

これらの点を注意して，機械語をメモリーに書き込み，実際に実行するとどのようになるか，を示したものが，**第6－3図**です。

7D19,7D1Aがそれぞれ00，00から02，01に変化しています。RLCAがSフラグに影響を与えなかったので，7D19は02になったわけです。一方，RLC　AはSフラグに影響を与えたので，7D19は01になったわけです。

さて，もう一つのプログラムは，RLD命令のテストプログラムです。これは，RLD命令を説明する本文中に例として説明したものを，実際にプログラムしてみたものです。



《第6－4図》
RLD命令を使ったテストプログラム

フローチャートを**第6－4図**に，アセンブラプログラムを**第6－5図**に，実行結果を**第6－6図**にまとめておきました。7D0D以下に書かれた16進数（10進数と見なすのも可）……012345，（メモリーには45－23－01の順に入っている）がプログラム実行後，123450（メモリーには，50－34－12と入っている）となったことがおわかりだと思います。

詳しくは，RLD命令の説明のところを参照してください。

```
     MACRO-80 3.3    09-MAY-79        PAGE    1
                              00100   ;        TEST PROGRAM ( RLD )
                              00200   .Z80
  0000'                       00300   ASEG
                              00400   ORG      7D00H
                              00500   ;
  7D00    AF                  00600            XOR     A              ; A=0
  7D01    21 7D0D             00700            LD      HL,DATA        ; HL=DATA
  7D04    ED 6F               00800            RLD
  7D06    23                  00900            INC     HL             ; HL=HL+1
  7D07    ED 6F               01000            RLD
  7D09    23                  01100            INC     HL             ; HL=HL+1
  7D0A    ED 6F               01200            RLD
  7D0C    C9                  01300            RET                    ; RETURN
                              01400   ;
  7D0D    45                  01500   DATA:    DB      45H            ; M(DATA)=45
  7D0E    23                  01600            DB      23H            ; M(DATA+1)=23
  7D0F    01                  01700            DB      01H            ; M(DATA+2)=01
                              01800            END
```

《第6－5図》　第6－4図のフローチャートをもとに，アセンブラプログラムしたもの

** BEFORE RUN

NO 1	7D00	AF
NO 2	7D01	21
NO 3	7D02	0D
NO 4	7D03	7D
NO 5	7D04	ED
NO 6	7D05	6F
NO 7	7D06	23
NO 8	7D07	ED
NO 9	7D08	6F
NO 10	7D09	23
NO 11	7D0A	ED
NO 12	7D0B	6F
NO 13	7D0C	C9
NO 14	7D0D	45
NO 15	7D0E	23
NO 16	7D0F	01

** AFFTER RUN

NO 1	7D00	AF
NO 2	7D01	21
NO 3	7D02	0D
NO 4	7D03	7D
NO 5	7D04	ED
NO 6	7D05	6F
NO 7	7D06	23
NO 8	7D07	ED
NO 9	7D08	6F
NO 10	7D09	23
NO 11	7D0A	ED
NO 12	7D0B	6F
NO 13	7D0C	C9
NO 14	7D0D	58
NO 15	7D0E	34
NO 16	7D0F	12

《第６－６図》 第６－５図の機械語を直接メモリーに書き込み，実行前と実行後の内容を比較したもの

7 1ビットだけに関わる命令

Ｚ８０では，レジスタあるいはメモリーの，一つのビットだけを操作したり，チェックしたりする命令があります。ここでは，そういった命令について，述べることにします。

レジスタまたはＨＬレジスタペアで指定されるメモリーのあるビットの状態を調べる命令

たとえば，レジスタＢの第３ビットがゼロかど

うか調べる場合は

BIT 3，B

という命令を使います。この命令を実行した場合に，もし，レジスタBの第3ビットがゼロならばZフラグが，1になります。そうでなければ，Zフラグはゼロになります。

したがって，この命令は，ゼロブランチ命令，あるいは非ゼロブランチ命令との組み合わせで用いるのが一般的です。

この命令を記号で表わす場合は，次のようにします。

Z← $\overline{B<3>}$

B＜3＞はレジスタBの第3ビットを表わしています。その上にバーが付いていますが，これは，ビットの内容を反転，つまり，ゼロならば1に，1ならばゼロにする，という意味です。したがって，レジスタBの第3ビットがゼロならば，Zフラグは1に，そうでなければ，ゼロになる，とい

《第7－1表》 レジスタ，またはHLレジスタペアで指定されるメモリーのあるビットをチェックしたり操作したりする命令の機械語の2ワード目を求める表（1ワード目はすべてCB）

上位4ビット	0	1	2	3	4	5	6	7
4	BIT 0，B	BIT 0，C	BIT 0，D	BIT 0，E	BIT 0，H	BIT 0，L	BIT 0,(HL)	BIT 0，A
5	BIT 2，B	BIT 2，C	BIT 2，D	BIT 2，E	BIT 2，H	BIT 2，L	BIT 2,(HL)	BIT 2，A
6	BIT 4，B	BIT 4，C	BIT 4，D	BIT 4，E	BIT 4，H	BIT 4，L	BIT 4,(HL)	BIT 4，A
7	BIT 6，B	BIT 6，C	BIT 6，D	BIT 6，E	BIT 6，H	BIT 6，L	BIT 6,(HL)	BIT 6，A
8	RES 0，B	RES 0，C	RES 0，D	RES 0，E	RES 0，H	RES 0，L	RES 0,(HL)	RES 0，A
9	RES 2，B	RES 2，C	RES 2，D	RES 2，E	RES 2，H	RES 2，L	RES 2,(HL)	RES 2，A
A	RES 4，B	RES 4，C	RES 4，D	RES 4，E	RES 4，H	RES 4，L	RES 4,(HL)	RES 4，A
B	RES 6，B	RES 6，C	RES 6，D	RES 6，E	RES 6，H	RES 6，L	RES 6,(HL)	RES 6，A
C	SET 0，B	SET 0，C	SET 0，D	SET 0，E	SET 0，H	SET 0，L	SET 0,(HL)	SET 0，A
D	SET 2，B	SET 2，C	SET 2，D	SET 2，E	SET 2，H	SET 2，L	SET 2,(HL)	SET 2，A
E	SET 4，B	SET 4，C	SET 4，D	SET 4，E	SET 4，H	SET 4，L	SET 4,(HL)	SET 4，A
F	SET 6，B	SET 6，C	SET 6，D	SET 6，E	SET 6，H	SET 6，L	SET 6,(HL)	SET 6，A
	0	1	2	3	4	5	6	7
	下位4ビット							

8	9	A	B	C	D	E	F	上位4ビット
BIT 1，B	BIT 1，C	BIT 1，D	BIT 1，E	BIT 1，H	BIT 1，L	BIT 1,(HL)	BIT 1，A	4
BIT 3，B	BIT 3，C	BIT 3，D	BIT 3，E	BIT 3，H	BIT 3，L	BIT 3,(HL)	BIT 3，A	5
BIT 5，B	BIT 5，C	BIT 5，D	BIT 5，E	BIT 5，H	BIT 5，L	BIT 5,(HL)	BIT 5，A	6
BIT 7，B	BIT 7，C	BIT 7，D	BIT 7，E	BIT 7，H	BIT 7，L	BIT 7,(HL)	BIT 7，A	7
RES 1，B	RES 1，C	RES 1，D	RES 1，E	RES 1，H	RES 1，L	RES 1,(HL)	RES 1，A	8
RES 3，B	RES 3，C	RES 3，D	RES 3，E	RES 3，H	RES 3，L	RES 3,(HL)	RES 3，A	9
RES 5，B	RES 5，C	RES 5，D	RES 5，E	RES 5，H	RES 5，L	RES 5,(HL)	RES 5，A	A
RES 7，B	RES 7，C	RES 7，D	RES 7，E	RES 7，H	RES 7，L	RES 7,(HL)	RES 7，A	B
SET 1，B	SET 1，C	SET 1，D	SET 1，E	SET 1，H	SET 1，L	SET 1,(HL)	SET 1，A	C
SET 3，B	SET 3，C	SET 3，D	SET 3，E	SET 3，H	SET 3，L	SET 3,(HL)	SET 3，A	D
SET 5，B	SET 5，C	SET 5，D	SET 5，E	SET 5，H	SET 5，L	SET 5,(HL)	SET 5，A	E
SET 7，B	SET 7，C	SET 7，D	SET 7，E	SET 7，H	SET 7，L	SET 7,(HL)	SET 7，A	F
8	9	A	B	C	D	E	F	
下位4ビット								

(例) BIT, 3, Hの機械語は，CB・5C

うことを表わすことになるわけです。

アセンブラ言語と機械語の対応は，命令の数が多いので，**第7－1表**にまとめておきました。

これらの命令は2ワード命令ですが，1ワード目は必ず，CBなので，**第7－1表**では2ワード目との対応が示してあります。

たとえば，

BIT　5，(HL)

という命令の場合，**第7－1表**より，6Eを知りその前に，CBをつけて

CB・6E

とすれば，これがこの命令の機械語というわけです。

これらの命令が各フラグに与える影響は，次の通りです。

S　：　どうなるかわからない

Z　：　指定したビットが1ならばゼロに，そうでなければ，ゼロになる

P/V：　どうなるかわからない

CY：　影響をうけず，前の状態を保持する

レジスタまたはHLレジスタペアで指定されるメモリーの，あるビットをゼロにする命令

たとえば，M(7D10)の第ゼロビットつまり最下位ビットをゼロにしたい場合は，HLレジスタペアに，7D10を入れた後に，

RES　0，(HL)

という命令を実行すればよいでしょう。RESはもちろん，RESETの意味です。

この命令の機械語は

CB・86

です。記号は次のようにしておきます。

M(HL)<0>←1

このような命令は全部で64種ありますが，そのアセンブラ言語と機械語の対応は，BIT命令と同じく，**第7－1表**にまとめておきました。すべて2ワード命令ですが，表では2ワード目の機械語との対応が示されています。1ワード目はどれもCBです。

たとえば，

RES　5，L

なら，その機械語は

CB・AD

というわけです。

これらの命令は，どのフラグにも影響は与えません。

レジスタまたはHLレジスタペアで指定されるメモリーの，あるビットを1にする命令

丁度，RES命令の逆の命令です。BLレジスタの第ゼロビットを1にする場合，つまり，

B<0>←1

アセンブラ言語で表わせば

SET　0，B

となります。そして，これの機械語は，やはり**第7－1表**に示したように，

CB・Cφ

となります。

そして，これらの命令は，どのフラグにも影響は与えません。

インデックスレジスタを用いて，メモリーを指定し，そのメモリーに対して，BIT，RES，SET命令を行なう場合

たとえば

$Z \leftarrow \overline{M(IX+12)<0>}$

という命令，つまり，インデックスレジスタIXに16進数12を加えたときにできる数を，アドレスとするようなメモリーの第ゼロビットの持つ情報の逆の値を，Zフラグへ送る命令は，アセンブラでは

BIT　0，(IX+12H)

と書きます。

この機械語は，**第7－2表**にも書いておきましたが

DD・CB・12・46

と，4ワードにまたがる長いものになります。

このうち，DD・CBはインデックスレジスタIXを用いるすべての場合に共通で，次の1ワードには，IXに加える16進数が入ります。そして，最後の1ワードは，BIT命令か，RES命令か，

《第7－2表》 インデックスレジスタを用いるＢＩＴ，ＲＥＳ，ＳＥＴ命令の機械語

		下位4ビット	
		6	E
上位4ビット	4	ＢＩＴ　0，～	ＢＩＴ　1，～
	5	ＢＩＴ　2，～	ＢＩＴ　3，～
	6	ＢＩＴ　4，～	ＢＩＴ　5，～
	7	ＢＩＴ　6，～	ＢＩＴ　7，～
	8	ＲＥＳ　0，～	ＲＥＳ　1，～
	9	ＲＥＳ　2，～	ＲＥＳ　3，～
	A	ＲＥＳ　4，～	ＲＥＳ　5，～
	B	ＲＥＳ　6，～	ＲＥＳ　7，～
	C	ＳＥＴ　0，～	ＳＥＴ　1，～
	D	ＳＥＴ　2，～	ＳＥＴ　3，～
	E	ＳＥＴ　4，～	ＳＥＴ　5，～
	F	ＳＥＴ　6，～	ＳＥＴ　7，～

例.

ＢＩＴ　0，（ＩＸ＋１２Ｈ）の機械語は

ＤＤ・ＣＢ・１２・４６

ＩＸを用いる場合はすべてこれ　　上の表より

ＳＥＴ　7，（ＩＹ＋３４Ｈ）の機械語は，

ＦＤ・ＣＢ・３４・ＦＥ

ＩＹを用いる場合はすべてこれ　　上の表より

ＳＥＴ命令，第何ビットを指定するかによって決まります。それは**第7－2表**にまとめた通りです。

Ｍ（ＩＹ＋３４）＜７＞←１

の場合のアセンブラ，機械語は

ＳＥＴ　７，（ＩＹ＋３４Ｈ）

ＦＤ・ＣＢ・３４・ＦＥ

となりますが，この，ＦＤ・ＣＢは，インデックスレジスタＩＹを用いるすべての場合に共通です。次の３４は，ＩＹに加える16進数です。最後のＦＥは，ＢＩＴ命令，ＲＥＳ命令，ＳＥＴ命令のうちどれか，また第何ビットを指定するかによって決まる値です。それについては**第7－2表**を参照してください。

これらの命令のうち，ＲＥＳ命令，ＳＥＴ命令は，どのフラグにも全く影響を与えません。ＢＩＴ命令が各フラグに与える影響は次の通りです。

Ｓ　：　どうなるかわからない
Ｚ　：　指定したビットが0ならば，Ｚ＝1，そうでなければ，Ｚ＝0
Ｐ/Ｖ：　どうなるかわからない
ＣＹ：　影響をうけない

RES，SET命令を使ったテストプログラム

これらの命令は特にむずかしいものではありませんが，確認の意味で**第7－1図**に示したフローチャートの働きをするプログラムを作ってみました。

これを実行すると，Ｍ(DATA)の第3ビットが1になり，Ｍ(DATA＋1)の第5ビットがゼロになるはずです。あらかじめ，Ｍ(DATA)をゼロに，

M(DATA＋1)をFFにしておけば，実行後，それぞれ

　０００００００００

→０００００１０００

　……〔０８〕

　１１１１１１１

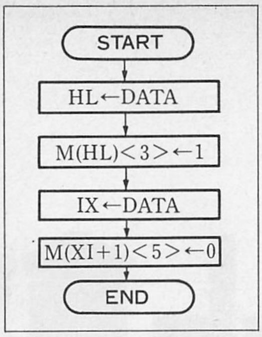
→１１０１１１１１

　……〔ＤＦ〕

となるはずです。

　これらのことを盛り込んで，アセンブラプログラムしたものが，**第７－２図**に表わしたものです。

　これをもとにして，７Ｄ００番地から直接機械語を書き込んだもの，そして実行した結果を示したものが**第７－３図**です。



《第７－１図》
ＲＥＳ，ＳＥＴ命令を使ったテストプログラム

```
MACRO-80 3.3    09-MAY-79       PAGE    1
                                00100   ;       TEST PROGRAM ( SET, RES )
                                00200    Z80
0000'                           00300   ASEG
                                00400   ORG     7D00H
                                00500   ;
7D00    21 7D0E                 00600           LD      HL,DATA         ; HL=DATA
7D03    CB DE                   00700           SET     3,(HL)          ; M(HL)<3>=1
7D05    DD 21 7D0E              00800           LD      IX,DATA         ; IX=DATA
7D09    DD CB 01 AE             00900           RES     5,(IX+1H)       ; M(IX+1)<5>=0
7D0D    C9                      01000           RET                     ; RETURN
                                01100   ;
7D0E    00                      01200   DATA.   DB      0H              ; M(DATA)=0
7D0F    FF                      01300           DB      11111111B       ; M(DATA+1)=11111111
                                01400           END
```

《第７－２図》　第７－１図のフローチャートをもとに，アセンブラプログラミングをしたもの

** BEFORE RUN

NO 1	7D00	21
NO 2	7D01	0E
NO 3	7D02	7D
NO 4	7D03	CB
NO 5	7D04	DE
NO 6	7D05	DD
NO 7	7D06	21
NO 8	7D07	0E
NO 9	7D08	7D
NO 10	7D09	DD
NO 11	7D0A	CB
NO 12	7D0B	01
NO 13	7D0C	AE
NO 14	7D0D	C9
NO 15	7D0E	00
NO 16	7D0F	FF

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	0E
NO 3	7D02	7D
NO 4	7D03	CB
NO 5	7D04	DE
NO 6	7D05	DD
NO 7	7D06	21
NO 8	7D07	0E
NO 9	7D08	7D
NO 10	7D09	DD
NO 11	7D0A	CB
NO 12	7D0B	01
NO 13	7D0C	AE
NO 14	7D0D	C9
NO 15	7D0E	08
NO 16	7D0F	DF

《第７－３図》　第７－２図のプログラムの機械語を直接メモリーに書き込み，実行前と実行後の内容を調べたもの

8　CALL命令とRET命令

ここでは，メインルーチンからサブルーチンを呼び出す命令……これをCALL命令といいますが，それと，サブルーチンからメインルーチンへ帰す命令……これをRET命令といいますが，この命令を説明します。

CALL命令，RET命令は大きく分けてそれぞれ二つの種類があります。一つは，無条件に，サブルーチンを呼び出したり，メインルーチンへ帰したりするものです。もう一つは，指定した条件を満たしている場合にのみサブルーチンを呼び出したり，メインルーチンへ帰したりするものです。

まずは無条件の場合について説明していきますが，ここで，どのようにしてサブルーチンを呼び出すのか，あるいはメインへ帰すのかを十分理解しておいてください。

無条件でサブルーチンを呼び出す命令

サブルーチンの先頭アドレスに，そのサブルーチンの働きにふさわしいシンボルを付けておくと便利です。今，掛け算のサブルーチンがあったとして，この先頭アドレスに，MULTというシンボルが付けられていたとします。このサブルーチンが必要なときは，次のように命令します。

```
CALL MULT
```

もし，シンボルが付けられていない場合は，その先頭アドレスを直接MULTに替えて書きます。たとえば，このサブルーチンの先頭アドレスが，7E00なら，

```
CALL 7E00
```

となります。

この場合の機械語は

```
CD・00・7E
```

です。CDは無条件CALLのコードで，00・7Eは，7E00という先頭アドレスから来ているものです。

実は，この命令を十分に使いこなすには，これだけの知識では不十分なのです。この命令が具体的には，どのような手順で実行されていくのかを理解する必要があります。

今とりあげた

```
CALL 7E00
```

を記号で書くと，次のようになるのです。

①M（SP−1）M（SP−2）←PC
②SP←SP−2
③PC←7E00

SPはスタックポインタ，PCはプログラムカウンタです。

スタックポインタが8200で，このCALL命令が7D00から書き込まれていたとすると，つまり

SP←8200

7D00	CD	CALL　7E00
7D01	00	
7D02	7E	
7D03	……	

というようになっていたとすると，命令実行後，PC，SP，M（81FF），M（81FE）は次のようになります。

M（81FF）←7D
M（81FE）←03
SP←81FE
PC←7E00

もう少し具体的に説明してみましょう。

プログラムが7D00　に来ると，そこに書かれている命令が3ワード命令だということがわかります。その段階で，プログラムカウンタPCには3が加えられ

PC←7D03

となります。

次に

M（SP−1）M（SP−2）←PC

が実行されますが，SPは8200　なので

M（81FF）←PCの上位8ビット
M（81FE）←PCの下位8ビット

となります。具体的には

M（81FF）←7D
M（81FE）←03

となります。次に

SP←SP−2

が実行されますので

SP←81FE

となります。そして最後に

PC←7E00

が実行されます。これが実行されるということは7E00へジャンプするということに他なりません。

以上のようにして，

CALL　7E00

は実行されます。おわかりいただけたと思いますが，M（81FF），M（81FE）に，サブルーチン終了後に帰るべきアドレスが保存されるわけです。

なお，この命令はどのフラグにも影響を与えません。

無条件で，メインルーチンへ帰すRET命令

この命令の機能を記号で表わすと次のようになります。

①SP←SP＋2
②PC←M（SP−1）M（SP−2）

一つ前に説明した，CALL命令の丁度反対をしていることが，おわかりだと思います。

具体的に考えてみましょう。CALL命令のときの例の続きを考えてみます。そこでは，メインルーチンから，7E00　から始まるサブルーチンへジャンプしましたが，その時点で

M（81FF）←7D
M（81FE）←03
SP←81FE

となっています。サブルーチンの中で，これらの内容を変えないのが普通ですから，サブルーチンの最後にこのRET命令を置けば，まず，

①SP←SP＋2　（81FE＋2）

より，SPは8200　となります。次に

②PC←M（SP−1）M（SP−2）

より，PCにはM（81FF），M（81FE）の内容，つまり，7D03　が入ります。PCに7D03が入るということは，7D03番地にジャンプすることに他なりませんから，これでメインルーチンへ帰ることができるわけです。

なお，アセンブラは

RET

の一言で，機械語は

C9

です。

また，この命令は，どのフラグにも影響を与えません。

条件つきＣＡＬＬ命令，ＲＥＴ命令

与えられた条件を満たした場合に，無条件のところで説明した手順で目的のアドレスへジャンプさせる命令です。

条件・アセンブラ言語・機械語を**第８－１表**にまとめておきました。

なお，フラグの条件の持つ意味については，条件ジャンプ命令のところを参照してください（特にＰＯ，ＰＥについては注意してください）。

これらの命令は，どのフラグにも影響は与えません。

《第８－１表》

条　件	条件つきＣＡＬＬ命令		条件つきＲＥＴ命令	
	アセンブラ	機　械　語	アセンブラ	機械語
Ｚ＝０　なら	ＣＡＬＬ　ＮＺ，７Ｅ００Ｈ	Ｃ４・００・７Ｅ	ＲＥＴ　ＮＺ	Ｃ０
Ｚ＝１　なら	ＣＡＬＬ　Ｚ，７Ｅ００Ｈ	ＣＣ・００・７Ｅ	ＲＥＴ　Ｚ	Ｃ８
ＣＹ＝０なら	ＣＡＬＬ　ＮＣ，７Ｅ００Ｈ	Ｄ４・００・７Ｅ	ＲＥＴ　ＮＣ	Ｄ０
ＣＹ＝１なら	ＣＡＬＬ　Ｃ，７Ｅ００Ｈ	ＤＣ・００・７Ｅ	ＲＥＴ　Ｃ	Ｄ８
Ｐ/Ｖ＝０なら	ＣＡＬＬ　ＰＯ，７Ｅ００Ｈ	Ｅ４・００・７Ｅ	ＲＥＴ　ＰＯ	Ｅ０
Ｐ/Ｖ＝１なら	ＣＡＬＬ　ＰＥ，７Ｅ００Ｈ	ＥＣ・００・７Ｅ	ＲＥＴ　ＰＥ	Ｅ８
Ｓ＝０　なら	ＣＡＬＬ　Ｐ，７Ｅ００Ｈ	Ｆ４・００・７Ｅ	ＲＥＴ　Ｐ	Ｆ０
Ｓ＝１　なら	ＣＡＬＬ　Ｍ，７Ｅ００Ｈ	ＦＣ・００・７Ｅ	ＲＥＴ　Ｍ	Ｆ８

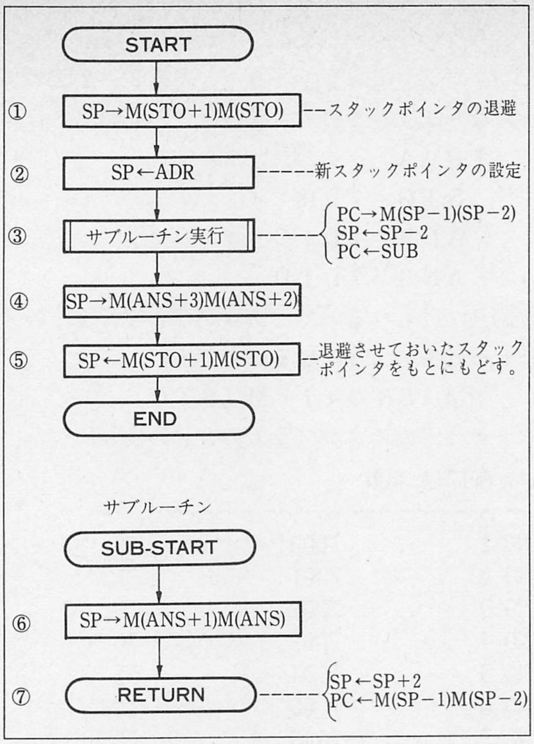
注．ここでは，サブルーチンの先頭アドレスは７Ｅ００としてあります。そこにシンボルがつけてあれば，アセンブラの７Ｅ００Ｈのかわりに，そのシンボルを書くことができます。

ＣＡＬＬ命令，ＲＥＴ命令を使ったテストプログラム

ＣＡＬＬ命令を使うと，スタックポインタＳＰやＭ（ＳＰ－１），Ｍ（ＳＰ－２）が具体的にどのように変化するのか，そして，その後ＲＥＴ命令を使うと，ＳＰがどのようになるのかを確認できるようなプログラムを作ってみました。

ただ，ここで注意しなければいけないのは，このテストプログラム自体がサブルーチンになっていることです。ですから，フローチャート上でメインプログラムの終りは，アセンブラプログラムのところでは，ＲＥＴ命令となっています。

また，ここでのメインを始める前に，以前のスタックポインタをＭ（ＳＴＯ＋１）Ｍ（ＳＴＯ）



《第８－１図》　ＣＡＬＬ命令，ＲＥＴ命令を使った場合に，ＳＰ，Ｍ（ＳＰ－１）Ｍ（ＳＰ－２）はどのように変化するかを確認するプログラム

に退避させている点にも注意してください。

第８－１図のフローチャートを少し説明してみましょう。

②のところで，ＳＰをあらたに，ＡＤＲとします。③で，ＣＡＬＬ命令を実行しますが，これは，

ＤＣ→Ｍ（ＳＰ－１）Ｍ（ＳＰ－２）

ＳＤ←ＳＰ－２

ＰＣ←ＳＵＢ

で，このとき

ＰＣ＝〔④の命令の先頭アドレス〕

ＳＰ＝ＡＤＲ

ですから，具体的には，

M(ADR-1)M(ADR-2)＝〔④の命令の先頭アドレス〕

ＳＰ←ＡＤＲ－２

ＰＣ←ＳＵＢ

となります。ＰＣ←ＳＵＢを実行すると⑥に行きますが，ここでこの時点でのＳＰの内容，つまり(ADR-２)はM(ANS＋１)M(ANS)に送られます。

この後の⑦で，ＲＥＴ　命令が実行されますが，これは

ＳＰ←ＳＤ＋２

ＰＣ←Ｍ（ＳＰ－１）Ｍ（ＳＰ－２）

のことで，この時点で，

ＳＰ＝（ＡＤＲ－２）

```
MACRO-80 3.3    09-MAY-79       PAGE    1
                         00100  ; REST PROGRAM ( CALL, RET )
                         00200  .Z80
0000'                    00300  ASEG
                         00400  ORG     7D00H
                         00500  ;
                         00600  ;--------- MAIN ----------------------------------------
7D00    ED 73 7D18       00700          LD      (STO),SP        ; M(SOT+1)M(SOT)=SP
7D04    31 7D1C          00800          LD      SP,ADR          ; SP=ADR
7D07    CD 7D13          00900          CALL    SUB             ; M(SP-1)M(SP-2)=PC :   SP=SP-2 : PC=SUB
7D0A    ED 73 7D1F       01000          LD      (ANS+2),SP      ; M(ANS+3)M(ANS+2)=SP
7D0E    ED 7B 7D18       01100          LD      SP,(STO)        ; SP=M(STO+1)M(STO)
7D12    C9               01200          RET                     ; MAIN END
                         01300  ;---------- SUB ----------------------------------------
7D13    ED 73 7D1D       01400  SUB.    LD      (ANS),SP        ; M(ANS+1)M(ANS)=SP
7D17    C9               01500          RET                     ; SP=SP+2 : PC=M(SP-1)M(SP-2)
                         01600  ;----------- SUB END ----------------------------------
7D18    00               01700  STO.    DB      0H              ; M(STO)=0
7D19    00               01800          DB      0H              ; M(STO+1)=0
7D1A    00               01900          DB      0H              ; M(ADR-2)=0
7D1B    00               02000          DB      0H              ; M(ADR-1)=0
7D1C    00               02100  ADR.    DB      0H              ; M(ADR)=0
7D1D    00               02200  ANS.    DB      0H              ; M(ANS)=0
7D1E    00               02300          DB      0H              ; M(ANS+1)=0
7D1F    00               02400          DB      0H              ; M(ANS+2)=0
7D20    00               02500          DB      0H              ; M(ANS+3)=0
                         02600          END
```

《第８－２図》　第８－１図のフローチャートをもとに，アセンブラプログラムをしたもの

なので，

SP←SP+2

を実行すると

SP=ADR

になります。したがって次の命令は

PC←M（ADR－1）M（ADR－2）

ということになります。M(ADR－1）M(ADR－2）には，④の先頭アドレスが入っていますから，これを実行するということは，④へジャンプするということになります。

④で，この時点のSPの値，つまりADRがM（ANS+3）M（ANS+2）に送られて，このプログラムは終ります。

このフローチャートを　7D00から，アセンブラプログラミングしてみると，**第8－2図**のようになります。ここで，④の命令の先頭アドレスは　7D0A。三つのシンボルは

STO=7D18

ADR=7D1C

ANS=7D1D

と割りつけられるので，プログラム実行後，各メモリーは次のようになるはずです。

M（STO+1）M（STO）

=〔前のプログラムのSPの値〕

** BEFORE RUN

NO 1	7D00	ED
NO 2	7D01	73
NO 3	7D02	18
NO 4	7D03	7D
NO 5	7D04	31
NO 6	7D05	1C
NO 7	7D06	7D
NO 8	7D07	CD
NO 9	7D08	13
NO 10	7D09	7D
NO 11	7D0A	ED
NO 12	7D0B	73
NO 13	7D0C	1F
NO 14	7D0D	7D
NO 15	7D0E	ED
NO 16	7D0F	7B
NO 17	7D10	18
NO 18	7D11	7D
NO 19	7D12	C9
NO 20	7D13	ED
NO 21	7D14	73
NO 22	7D15	1D
NO 23	7D16	7D
NO 24	7D17	C9
NO 25	7D18	00
NO 26	7D19	00
NO 27	7D1A	00
NO 28	7D1B	00
NO 29	7D1C	00
NO 30	7D1D	00
NO 31	7D1E	00
NO 32	7D1F	00
NO 33	7D20	00

** AFFTER RUN

NO 1	7D00	ED
NO 2	7D01	73
NO 3	7D02	18
NO 4	7D03	7D
NO 5	7D04	31
NO 6	7D05	1C
NO 7	7D06	7D
NO 8	7D07	CD
NO 9	7D08	13
NO 10	7D09	7D
NO 11	7D0A	ED
NO 12	7D0B	73
NO 13	7D0C	1F
NO 14	7D0D	7D
NO 15	7D0E	ED
NO 16	7D0F	7B
NO 17	7D10	18
NO 18	7D11	7D
NO 19	7D12	C9
NO 20	7D13	ED
NO 21	7D14	73
NO 22	7D15	1D
NO 23	7D16	7D
NO 24	7D17	C9
NO 25	7D18	2E
NO 26	7D19	B8
NO 27	7D1A	0A
NO 28	7D1B	7D
NO 29	7D1C	00
NO 30	7D1D	1A
NO 31	7D1E	7D
NO 32	7D1F	1C
NO 33	7D20	7D

《第8－3図》　第8－2図から与えられた機械語を直接メモリーに書き込み，実行前と実行後の内容を比較したもの

M（ADR－2）＝0A
M（ADR－1）＝7D
M（ANS）＝1A
M（ANS＋1）＝7D
M（ANS＋2）＝1C
M（ANS＋3）＝7D

第8－2図から与えられた機械語を，7D00番地から書き込んで実行すると，実際に上のようになることか確認されます。それを**第8－3図**にまとめておきました。

9 レジスタやCPUをコントロールする命令

ここでは次のような命令をとりあげます。

CPL, NEG, CCF, SCF, NOP,
HALT, DAA, DI, EI, IM0,
IM 1, IM 2,

これらが，どのような命令であるかを，一つ一つ説明していきます。

CPL命令

この命令は，レジスタAの8個のビットの持つ情報を，すべて反転させる命令です。たとえば，レジスタAが

10110001

となっている場合に，この命令を実行すると，

01001110

となります。

この命令を記号で書くと

$A \leftarrow \overline{A}$

となります。

その機械語は

2F

です。

また，この命令は，次のフラグには影響を与えません。

S, Z, P／V, CY

NEG命令

この命令を記号で表わすと，次のようになります。

$A \leftarrow -A$

つまり，レジスタAの8ビットが，正負の数を

表わしていると考えたときに，その符号を反転させる命令です。

たとえば，レジスタAが16進数で

FF ……………（－1のこと）

のときに，この命令を実行すると，

01 ……………（＋1のこと）

となります。

この命令の機械語は，

ED・44

です。

この命令が，各フラグに与える影響は，次の通りです。

S ： 命令実行後，レジスタAの最上位ビットが1ならば，S＝1，そうでなければ，S＝0

Z ： 命令実行後，レジスタAがゼロならば，Z＝1，そうでなければ，Z＝0

P/V： 命令実行前に，レジスタAが16進数80（10進数の－256に当る）であった場合に，P／V＝1，そうでなければP／V＝0

CY： 命令実行前に，レジスタAがゼロでなければ，CY＝1，ゼロならば，CY＝0

CCF命令

この命令は，キャリーフラグの持つ情報を反転させる命令です。

つまり，この命令実行前に，CY＝1 ならば，実行後は CY＝0，になるというものです。

この命令を記号で書くと

$CY \leftarrow \overline{CY}$

となります。機械語は

3F

です。

この命令は，CY以外の，S，Z，P／Vのフラグには影響を与えません。

SCF命令

この命令は，キャリーフラグを1にする命令です。記号で書くと

CY←1

となります。その機械語は

37

です。

この命令は，CY以外の S，Z，P／Vフラグには影響を与えません。

ここで参考までに一言

CY←0

という命令はありませんが，これは色々な命令で実現できます。たとえば，

OR A………（A←A∨A）

AND A……（A←A∧A）

ついでに A←0 としたければ

XOR A …（A←A∀A）

SUB A …（A←A－A）

NOP命令

この命令は，何もしないという命令です。すでにできあがってしまったプログラムの1部を削除したいときなどに，埋め合わせとして使われます。

機械語は，

00

です。もちろんどのフラグにも影響は与えません。

HALT命令

HALTの持つ意味通り，プログラムを停止させる命令です。

各フラグ等すべてに影響は与えず，この命令の書かれている番地で，CPUはストップします。ただし，割り込みを受けつけることはできます。

この命令の機械語は，

76

です。

DAA命令

DAAは，DECIMAL ADJUST — ACCUMULATOR の略です。この命令を足し算命令（ADD，ADC，INC）や，引き算命令（SUB，SBC，DEC，NEG）の次に配置すると，これらの演算命令がBCDコードで表わした10進数に対して有効なものになります。具体例をテストプログラムのところでとりあげてあるので，そちらを参照してください。

この命令の機械語は

27

です。

また，各フラグに与える影響は次の通りです。

S： 命令実行後，レジスタAの最上位ビットが1ならば，S＝1，そうでなければ，S＝0

Z： 命令実行後，レジスタAがゼロならば，Z＝1，そうでなければZ＝0

P/V： 命令実行後，レジスタAの8個のビットのうち，1になっているものの数が偶数ならば，P／V＝1，そうでなければP／V＝0

CY： 上位4ビットを10進数に補正した時点で，10進数として上の桁に影響を与えるような場合，CY＝1，そうでなければ，CY＝0

DI命令

DIは，DISABLE INTERRUPTS の略です。この命令を実行すると，マスク可能な割り込みをうけつけなくなります（インタラプト・エナブル・フリップ・フロップ……IFF1，IFF2がリセットされます）。

この命令の機械語は

F3

です。

フラグその他に影響を与えることはありません。

EI命令

EIは ENABLE INTERRUPTS の略です。この命令を実行するとマスク可能な割り込みが可能になります（インタラプト・エナブル・フリップーフロップーIFF1，IFF2，がセットされます)。もちろん，割り込み実行中には，他の割り込みは禁止されます。

この命令の機械語は

FD

です。

フラグその他へ影響を与えることはありません。

IM0, IM1, IM2

これらの命令は，それぞれ，割り込みのモードをゼロ，1，2 にする命令です。

割り込みに関しては，ハードウェアの十分な知識も必要になるので，簡単に説明することができませんが，モード0，モード1，モード2は次のようになっています。

モード0： 8080系と同じで，いかなる命令もうけつける。

モード1： プログラムカウンタの内容を，スタックポインタを使って退避させたあと，0038番地へジャンプする。

モード2： プログラムカウンタの内容を，スタックポインタを使って退避させたあと，インタラプト・ページ・アドレス・レジスタ（I）などで指定されるアドレスへジャンプする。

それぞれ，機械語は次のようになっています。

IM 0 ………ED・46
IM 1 ………ED・56
IM 2 ………ED・5E

もちろん，これらの命令は，どのフラグにも影響は与えません。

RETI

RETIは，RETURN FROM INTERRUPT の略です。この命令は，サブルーチンの最後にRET命令を置くのと同じ要領で，一つの割に込みに対応して作られたプログ

ラムの最後に置かれます。その働きも，サブルーチンにおけるRET命令と同じです。

また 次に処理しなければならない割り込みの要求が来ていれば，それの受け入れを許可します。

この命令の機械語は

ED・4D

です。

もちろん，どのフラグにも影響は与えません。

RETN命令

RETNは，RETURN FROM NON MASKABLE INTERRUPTの略です。

マスクされない割り込みに対応させたプログラムの最後に置く命令です。普通のプログラム中のサブルーチンの最後に使うRET命令に似ています。

この命令は具体的に次のように実行されます。

○SP←SP＋2
○IFF1←IFF2
○PC←M（SP−1）M（SP−2）

なお，IFF1，IFF2は，インタラプト・エナブル・フリップ・フロップのことです。

この命令の機械語は

ED・45

です。

もちろん，どのフラグにも影響は与えません。

RST命令

この命令を実行することによって，次の番地へジャンプすることができます。このとき，それまでのプログラムカウンタの内容は，スタック領域に保存されます。

0000
0008
0010
0018
0020
0028
0030
0038

この命令は一語命令なので，一つのシステムを作るときに，最も頻繁に用いるプログラムやサブルーチンなどを上の番地から書き込むようにしておくと便利です。

記号，アセンブラ，機械語の順でまとめると次のようになります。

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0000
RST 0 ………………… C7

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0008
RST 8 ………………… CF

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0010
RST 10H ………………… D7

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0018
RST 18H ………………… DF

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0020
RST 20H …………………E7

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0028
RST 28H ………………… EF

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0030
RST 30H ………………… F7

SP←SP−2，M(SP−1)M(SP−2)←PC
PC←0038
RST 38H………………… FF

DAA命令を使ったテストプログラム

ＢＣＤコードで表わされた10進数４桁どうしの足し算のプログラムを作ってみました。機能や考え方は**第９－１図**のフローチャートに見る通りです。テストプログラムということで，インデックスレジスタＩＸを作ってみましたが，特にこだわる必要はないでしょう。

第９－１図をもとに，アセンブラプログラミングしたものが，**第９－２図**です。

ここでは，6789＋1234　の計算をさせるようにプログラムしてあります。したがって

　　Ｍ（ＤＡＴＡ）＝８９
　　Ｍ（ＤＡＴＡ＋１）＝６７
　　Ｍ（ＤＡＴＡ＋２）＝３４
　　Ｍ（ＤＡＴＡ＋３）＝１２

とします。所定通りの働きをすれば

　　6789＋1234＝8023

より，

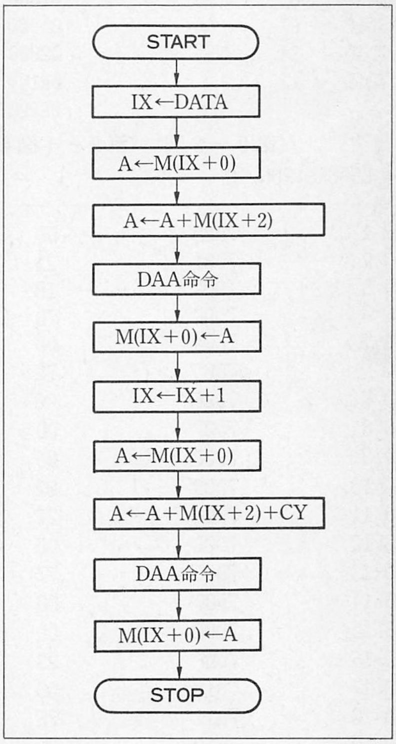
　　Ｍ（ＤＡＴＡ）＝２３
　　Ｍ（ＤＡＴＡ＋１）＝８０

になるはずです。

第９－２図の機械語を直接メモリーに入れ，実行前と実行後の変化をみたものが**第９－３図**です。実行後

　　Ｍ（ＤＡＴＡ）＝Ｍ（７Ｄ１Ｂ）＝２３
　　Ｍ（ＤＡＴＡ＋１）＝Ｍ（７Ｄ１Ｃ）＝８０

になっているのが確認できます。



《第９－１図》
Ｍ（ＤＡＴＡ＋１）Ｍ（ＤＡＴＡ）
←Ｍ（ＤＡＴＡ＋１）Ｍ（ＤＡＴＡ）
＋Ｍ（ＤＡＴＡ＋３）Ｍ（ＤＡＴＡ＋２）
ただし，10進数どうしの計算

```
     MACRO-80 3.3    09-MAY-79        PAGE    1
                               00100  ;        TEST PROGRAM ( DAA )
                               00200   Z80
 0000'                         00300  ASEG
                               00400  ORG      7D00H
                               00500  ,
 7D00    DD 21 7D1B            00600           LD       IX,DATA          ; IX=DATA
 7D04    DD 7E 00              00700           LD       A,(IX+0)         ; A=M(IX+0)
 7D07    DD 86 02              00800           ADD      A,(IX+2)         ; A=A+M(IX+2)
 7D0A    27                    00900           DAA
 7D0B    DD 77 00              01000           LD       (IX+0),A         ; M(IX+0)=A
 7D0E    DD 23                 01100           INC      IX               ; IX=IX+1
 7D10    DD 7E 00              01200           LD       A,(IX+0)         ; A=M(IX+0)
 7D13    DD 8E 02              01300           ADC      A,(IX+2)         ; A=A+M(IX+2)+CY
 7D16    27                    01400           DAA
 7D17    DD 77 00              01500           LD       (IX+0),A         ; M(IX+0)=A
 7D1A    C9                    01600           RET                       ; RETURN
                               01700  ;
 7D1B    89                    01800  DATA.    DB       89H              ; M(DATA)=89
 7D1C    67                    01900           DB       67H              ; M(DATA+1)=67
 7D1D    34                    02000           DB       34H              ; M(DATA+2)=34
 7D1E    12                    02100           DB       12H              ; M(DATA+3)=12
                               02200           END
```

《第9－2図》 第9－1図をもとに，アセンブラプログラミングしたもの

** BEFORE RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	1B
NO 4	7D03	7D
NO 5	7D04	DD
NO 6	7D05	7E
NO 7	7D06	00
NO 8	7D07	DD
NO 9	7D08	86
NO 10	7D09	02
NO 11	7D0A	27
NO 12	7D0B	DD
NO 13	7D0C	77
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	23
NO 17	7D10	DD
NO 18	7D11	7E
NO 19	7D12	00
NO 20	7D13	DD
NO 21	7D14	8E
NO 22	7D15	02
NO 23	7D16	27
NO 24	7D17	DD
NO 25	7D18	77
NO 26	7D19	00
NO 27	7D1A	C9
NO 28	7D1B	89
NO 29	7D1C	67
NO 30	7D1D	34
NO 31	7D1E	12

** AFFTER RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	1B
NO 4	7D03	7D
NO 5	7D04	DD
NO 6	7D05	7E
NO 7	7D06	00
NO 8	7D07	DD
NO 9	7D08	86
NO 10	7D09	02
NO 11	7D0A	27
NO 12	7D0B	DD
NO 13	7D0C	77
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	23
NO 17	7D10	DD
NO 18	7D11	7E
NO 19	7D12	00
NO 20	7D13	DD
NO 21	7D14	8E
NO 22	7D15	02
NO 23	7D16	27
NO 24	7D17	DD
NO 25	7D18	77
NO 26	7D19	00
NO 27	7D1A	C9
NO 28	7D1B	23
NO 29	7D1C	80
NO 30	7D1D	34
NO 31	7D1E	12

《第9－3図》 第9－2図の機械語を直接メモリーに入れ，実行前と実行後の変化をみたもの

10 入出力に関する命令

Z80では，入出力のポートが256個用意されています。したがって，最大256個の入出力装置を制御することが可能です。ここで説明する入出力命令は，これらの入出力装置のうち，どれから入力を得，どれへ出力するのかを指定する命令です。

直接，入力ポートを指定するＩＮ命令

この場合，入力装置からの情報はレジスタAにだけ送られます。もし，入力ポートが２３Ｈであれば，記号は次のようになります。

　　A←I/O（２３H）

アセンブラ，機械語は，

　　ＩＮ　A，（２３H）………DB・23

となります。

各フラグに影響を与えることはありません。

入力ポートをレジスタＣを使って指定するＩＮ命令

この場合，入力装置からの情報は，七つあるレジスタのうち，指定する一つに送られます。

記号，アセンブラ，機械語の順にまとめると次のようになります。

B←I/O（C）

　　ＩＮ　B，（C）……………ED・40

C←I/O（C）

　　ＩＮ　C，（C）……………ED・48

D←I/O（C）

　　ＩＮ　D，（C）……………ED・50

E←I/O（C）

　　ＩＮ　E，（C）……………ED・58

H←I/O（C）

　　ＩＮ　H，（C）……………ED・60

L←I/O（C）

　　ＩＮ　L，（C）……………ED・68

L←I/O（C）

　　ＩＮ　A，（C）……………ED・78

これらの命令が各フラグに与える影響は次の通りです。

S　：　送られてきたデータの最上位ビットが１ならば，S＝１，そうでなければ，

S＝0

Z　：　送られてきたデータがゼロならば，Z＝1，そうでなければ，Z＝0

P/V：　送られてきた情報をビット単位で考えた場合，1になっているビットの数が偶数ならば，P/V＝1，そうでなければ，P/V＝0

CY：　影響はうけない

直接，出力ポートを指定するOUT命令

256個ある出力ポートのうち，どこかのポートを指定し，そこにレジスタAの内容を送るという命令です。ハードウェアとの関わりが大きいのですが，あるポートにある数を出力すると，ある入出力装置の電源がONになる，といったようなことができます。後で，具体例を示しますが，TRS−80システムでは，FFポートに，第2ビットが1である8ビットの情報を送ると，そこに継いだカセットテープレコーダのモータがONになるようになっています。また同じポートに，第2ビットがゼロである8ビットの情報を送ると，そのモータがOFFになるようになっています。

出力ポートをFFに設定し，そこへレジスタAの内容を送る命令を記号で書くと，

I/O（FF）←A

となります。上の例の場合，

A<2>←1，または0

としてから，この命令を実行すれば目的は果たせます。

この場合，アセンブラ，機械語は次のようになります。

OUT　（0FFH），A

…………D3・FF

もちろん，この命令は，どのフラグにも影響を与えません。

出力ポートをレジスタCを使って指定するOUT命令

この場合，七つあるレジスタのどれか一つを選び，そこに入っている内容を，指定する出力ポートに送ることができます。

記号，アセンブラ，機械語の順でまとめると次のようになります。

I/O（C）←B
OUT　（C），B
……………………ED・41

I/O（C）←C
OUT　（C），C
……………………ED・49

I/O（C）←D
OUT　（C），D
……………………ED・51

I/O（C）←E
OUT　（C），E
……………………ED・59

I/O（C）←H
OUT　（C），H
……………………ED・61

I/O（C）←L
OUT　（C），L
……………………ED・69

I/O（C）←A
OUT　（C），A
……………………ED・79

これらの命令は，どのフラグにも影響を与えません。

OUT命令を使った
テストプログラム

ＯＵＴ命令をテストしてみる場合，困った問題があります。ハードウエア扱きに，この命令をテストすることはできないからです。

そこで，ここでは，Ｚ80を用いているパーソナルコンピュータ TRS-80 システムを例にとってみることにしました。

なるべく基本的なもの，ということで，TRS-80に継ぐカセットテープレコーダのモータをコントロールするプログラムを考えてみることにしましょう。

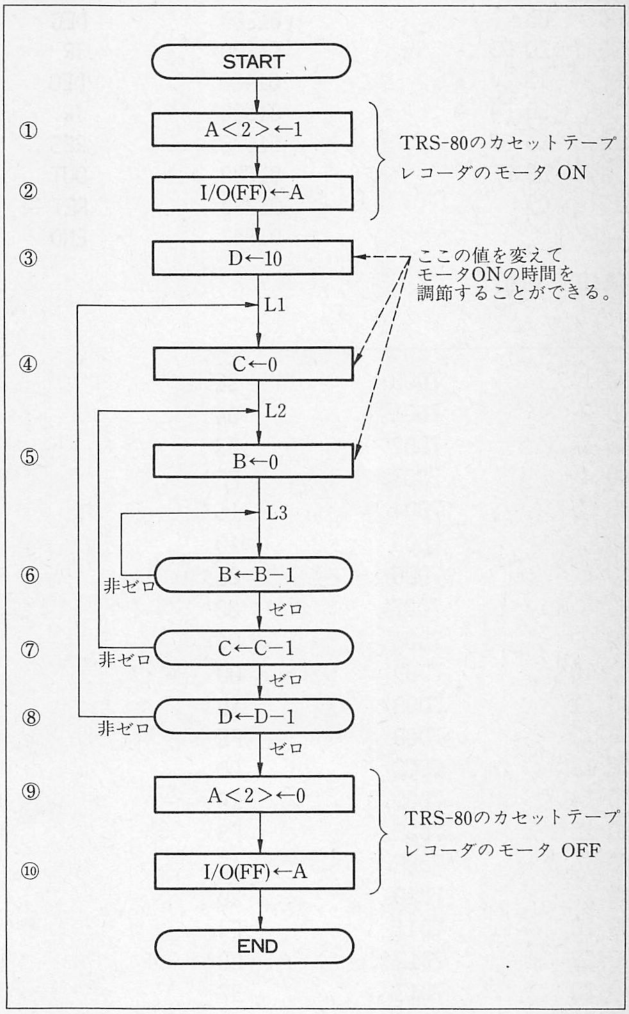
TRS-80では，出力ポートを 255（16進数でＦＦ）に指定し，そこに第２ビットが１である８ビットの情報を送ってやると，カセットテープレコーダのモータをＯＮにするリレーが閉じるようになっています。

また，出力ポートを 255（16進数でＦＦ）に指定し，そこに第２ビットがゼロである８ビットの情報を送ってやると，リレーが開くようになっています。

そこで，**第10－１図**のフローチャートに示されたようなプログラムを作れば，ある時間だけテープレコーダは動くはずです。

つまり，図中，①②で，テープレコーダは動き出し，③～⑧で条件が満たされるまでループして，⑨⑩で，テープレコーダは停止します。

③④⑤で各レジスタに入れる値を変えれば，テープレコーダが動いている時間が制



《第10－１図》　ＯＵＴ命令のテストプログラム（ＴＲＳ－80のカセットテープレコーダのモータＯＮ，ＯＦＦ制御）

御できます。これをもとに，アセンブラプログラムしたものが，**第10－2図**です。

第10－3図は，この**第10－2図**の機械語を直接メモリーに入れてみたところです。

ＴＲＳ－80に，マニュアルで指定された通りテープレコーダを接続してから，このプログラムを実行すると，約8秒間，テープレコーダは動きます。

```
    MACRO-80 3.3    09-MAY-79    PAGE    1
                            ;         . TEST PROGRAM ( OUT )
                            .Z80
0000'                       ASEG
                            ORG     7D00H
                            ;
7D00    3E 04                       LD      A,4             ; A<2>=1
7D02    D3 FF                       OUT     (0FFH),A        ; I/O(FF)=A
7D04    16 10                       LD      D,10H           ; D=10
7D06    0E 00               L1:     LD      C,0             ; C=0
7D08    06 00               L2:     LD      B,0             ; B=0
7D0A    10 FE               L3:     DJNZ    L3              ; B=B-1 : IF B<>0 GOTO L3
7D0C    0D                          DEC     C               ; C=C-1
7D0D    20 F9                       JR      NZ,L2           ; IF C<>0 GOTO L2
7D0F    15                          DEC     D               ; D=D-1
7D10    20 F4                       JR      NZ,L1           ; IF D<>0 GOTO L1
7D12    CB 97                       RES     2,A             ; A<2>=0
7D14    D3 FF                       OUT     (0FFH),A        ; I/O(FF)=A
7D16    C9                          RET                     ; RETURN
                                    END
```

《第10－2図》

```
---------------------------------------
 NO 1           7D00          3E
 NO 2           7D01          04
 NO 3           7D02          D3
 NO 4           7D03          FF
 NO 5           7D04          16
 NO 6           7D05          10
 NO 7           7D06          0E
 NO 8           7D07          00
 NO 9           7D08          06
 NO 10          7D09          00
 NO 11          7D0A          10
 NO 12          7D0B          FE
 NO 13          7D0C          0D
 NO 14          7D0D          20
 NO 15          7D0E          F9
 NO 16          7D0F          15
 NO 17          7D10          20
 NO 18          7D11          F4
 NO 19          7D12          CB
 NO 20          7D13          97
 NO 21          7D14          D3
 NO 22          7D15          FF
 NO 23          7D16          C9
---------------------------------------
```

《第10－3図》

11 一つのデータブロックを対象とした命令グループ

連続したメモリーに入っているデータを，連続的に対象とするときに便利な命令をここにまとめてみました。これらの命令には次のようなものがあります。

1，ロード命令系　LDI, LDD

2，比較命令系　CPI, CPD

3，入出力命令系　INI, IND, OUTI, OUTD,

一つ一つ説明していくことにしましょう。

ロード命令系

ＬＤＩ命令とＬＤＤ命令がありますが，それぞれを記号で表わすと次のようになります。

ＬＤＩ命令の場合。

①M(ＤＥ)←M(ＨＬ)

②ＤＥ←ＤＥ＋1

③ＨＬ←ＨＬ＋1

④ＢＣ←ＢＣ－1

ＬＤＤ命令の場合

①M(ＤＥ)←M(ＨＬ)

②ＤＥ←ＤＥ－1

③ＨＬ←ＨＬ－1

④ＢＣ←ＢＣ－1

どちらの命令の場合も，S，Z，CYフラグには影響は与えませんが，P/Vフラグに影響を与えます。BC←B－1　を実行後，BC≠0　ならばP/V＝1, BC＝0　ならば P/V＝0　となります。

したがって，1ブロック内のデータ数をレジスタペアBCに入れておき，これらの命令を実行するたびに，P/Vフラグをチェックすれば，必要回数命令が実行されたかどうか確認することができます。

それぞれの機械語は次の通りです。

ＬＤＩ……………………………ＥＤ・Ａ0

ＬＤＤ……………………………ＥＤ・Ａ8

比較命令系

ＣＰＩ命令と，ＣＰＤ命令とがあります。それぞれを記号で表わすと次のようになります。

ＣＰＩ命令の場合

①Null←Ａ－M（ＨＬ）

②ＨＬ←ＨＬ＋１

③ＢＣ←ＢＣ－１

ＣＰＤ命令の場合

①Null←Ａ－Ｍ（ＨＬ）

②ＨＬ←ＨＬ－１

③ＢＣ←ＢＣ－１

これらの命令が各フラグに与える影響は次の通りです。

S ： Ａ－Ｍ（ＨＬ）の結果が負ならば，つまり，第７ビットが１ならば，Ｓ＝１，そうでなければ，Ｓ＝０

Z ： ＡとＭ（ＨＬ）の内容が同じなら，Ｚ＝１，そうでなければ，Ｚ＝０

P/V： 命令実行後，ＢＣ＝０　ならば，P/V＝０，そうでなければ，P/V＝１

CY： 影響はうけない

以上のことからわかるように，ブロック内での最大値や最小値を見つけ出すときなどに便利な命令です。その点については，テストプログラムのところを参照してください。

機械語は次の通りです。

CPI……………………………ED・A1

CPD……………………………ED・A9

入出力命令系

ＩＮＩ命令，ＩＮＤ命令，ＯＵＴＩ命令，ＯＵＴＤ命令の四つがあります。これらの命令を記号で表わすと次のようになります。

ＩＮＩ命令の場合。

①Ｍ（ＨＬ）←Ｉ／Ｏ（Ｃ）

②ＨＬ←ＨＬ＋１

③Ｂ←Ｂ－１

ＩＮＤ命令の場合

①Ｍ（ＨＬ）←Ｉ／Ｏ（Ｃ）

②ＨＬ←ＨＬ－１

③Ｂ←Ｂ－１

ＯＵＴＩ命令の場合

①Ｉ／Ｏ（Ｃ）←Ｍ（ＨＬ）

②ＨＬ←ＨＬ＋１

③Ｂ←Ｂ－１

ＯＵＴＤ命令の場合

①Ｉ／Ｏ（Ｃ）←Ｍ（ＨＬ）

②ＨＬ←ＨＬ－１

③Ｂ←Ｂ－１

どの命令も各フラグに与える影響は同じで，次のようになっています。

S ： 不定

Z ： 命令実行後，Ｂがゼロならば，Ｚ＝１，そうでなければ，Ｚ＝０

P/V： 不定

CY： 影響はうけない

以上からわかるように，データを連続的に入力または出力するときに便利な命令です。

機械語との対応は次の通りです。

INI……………………………ED・A2

IND……………………………ED・AA

OUTI…………………………ED・A3

OUTD…………………………ED・AB

CPI命令を使ったテストプログラム

（最大値を求めるプログラム）

ＣＰＩ命令を使って最大値を求めるプログラムを考えてみました。考え方は**第11－1図**のフローチャートに見る通りです。

ここでは，８ビットのデータは正負を考慮して表現されているものとしてあります。ただし，オーバーフローが起きる場合には，正しい結果が得られません。

考え方の上でのポイントは，①でレジスタAに８０（10進数－128）を入れておくこと，④でＭ（ＨＬ）の内容をDに入れておくこと，の二つでしょう。

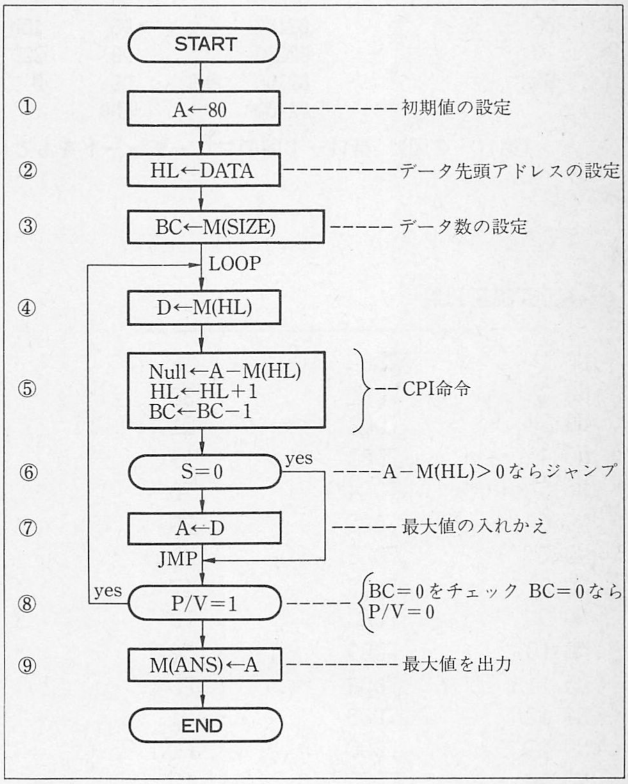
これをもとに，アセンブラプログラミングしたものが**第11－２図**です。データは10進数に直せば，－１，－２，４，15，32になるような五つの16進数としてあります。

M(SIZE＋１)M(SIZE)に５，つまり，０００５を入れ，M(DATA)以降にFF, FE, ０4，０F，２０，を入れ，M(ANS)にゼロを入れておきます。

これらのことをもとにして，７D00以降のメモリに直接，機械語，データを入れたものが**第11－３図**の BEFORE RUN の表に当ります。これを実行した後のメモリーの様子が，AFFTOR RUN の表に当ります。M(７D1E)に五つのデータの内，最大の２０が入っているのが確認できます。

もう１回実行してみましょう。M(ANS)にあたる７D1Eを００にして，M(７D1C)に，このプログラムであつかえる最大の数７F（＋127）を入れてみましょう。

実行前，実行後のメモリーの内容を**第11－４図**に示しておきました。M(７D1C)が，７Fになって，プログラムが所定の働きをしたことが確認できます。



《第11－１図》　ＣＰＩ命令を使ったテストプログラム
——最大値を求めるプログラム

```
MACRO-80 3.3    09-MAY-79        PAGE    1
                         00100  ;        TEST PROGRAM ( CPI )
                         00200  .Z80
0000'                    00300  ASEG
                         00400  ORG      7D00H
                         00500  ;
7D00   3E 80             00600           LD       A,80H          ; A=80
7D02   21 7D19           00700           LD       HL,DATA        ; HL=DATA
7D05   ED 4B 7D17        00800           LD       BC,(SIZE)      ; BC=M(SIZE+1)M(SIZE)
7D09   56                00900  LOOP:    LD       D,(HL)         ; D=M(HL)
7D0A   ED A1             01000           CPI                     ; NULL=A-M(HL) : HL=HL+1 : BC=BC-1
7D0C   F2 7D10           01100           JP       P,JMP          ; IF S=0 GOTO JMP
7D0F   7A                01200           LD       A,D            ; A=D
7D10   EA 7D09           01300  JMP:     JP       PE,LOOP        ; IF P/V=1 GOTO LOOP
7D13   32 7D1E           01400           LD       (ANS),A        ; M(ANS)=A
7D16   C9                01500           RET                     ; RETURN
                         01600  ;
7D17   05                01700  SIZE:    DB       5              ; M(SIZE)=5
7D18   00                01750           DB       0              ; M(SIZE+1)=0
7D19   FF                01800  DATA:    DB       -1D            ; M(DATA)=-1
7D1A   FE                01900           DB       -2D            ; M(DATA+1)=-2
7D1B   04                02000           DB       4              ; M(DATA+2)=4
7D1C   0F                02100           DB       15D            ; M(DATA+3)=15
7D1D   20                02200           DB       32D            ; M(DATA+4)=32
7D1E   00                02300  ANS:     DB       0              ; M(ANS)=0
                         02400           END
```

《第11－2図》 第11－1図のフローチャートをもとに，アセンブラプログラミングしたもの

```
** BEFORE RUN
--------------------------------------------
NO 1        7D00      3E
NO 2        7D01      80
NO 3        7D02      21
NO 4        7D03      19
NO 5        7D04      7D
NO 6        7D05      ED
NO 7        7D06      4B
NO 8        7D07      17
NO 9        7D08      7D
NO 10       7D09      56
NO 11       7D0A      ED
NO 12       7D0B      A1
NO 13       7D0C      F2
NO 14       7D0D      10
NO 15       7D0E      7D
NO 16       7D0F      7A
NO 17       7D10      EA
NO 18       7D11      09
NO 19       7D12      7D
NO 20       7D13      32
NO 21       7D14      1E
NO 22       7D15      7D
NO 23       7D16      C9
NO 24       7D17      05
NO 25       7D18      00
NO 26       7D19      FF
NO 27       7D1A      FE
NO 28       7D1B      04
NO 29       7D1C      0F
NO 30       7D1D      20
NO 31       7D1E      00
--------------------------------------------
```

《第11－3図①》 第11－2図の機械語を直接メモリーに入れ，プログラム実行前後のメモリーの内容を示したもの

```
** AFFTER RUN
----------------------------------------
NO 1        7D00        3E
NO 2        7D01        80
NO 3        7D02        21
NO 4        7D03        19
NO 5        7D04        7D
NO 6        7D05        ED
NO 7        7D06        4B
NO 8        7D07        17
NO 9        7D08        7D
NO 10       7D09        56
NO 11       7D0A        ED
NO 12       7D0B        A1
NO 13       7D0C        F2
NO 14       7D0D        10
NO 15       7D0E        7D
NO 16       7D0F        7A
NO 17       7D10        EA
NO 18       7D11        09
NO 19       7D12        7D
NO 20       7D13        32
NO 21       7D14        1E
NO 22       7D15        7D
NO 23       7D16        C9
NO 24       7D17        05
NO 25       7D18        00
NO 26       7D19        FF
NO 27       7D1A        FE
NO 28       7D1B        04
NO 29       7D1C        0F
NO 30       7D1D        20
NO 31       7D1E        20
----------------------------------------
```

《第11－3図②》 第11－2図の機械語を直接メモリーに入れ，プログラム実行前後のメモリーの内容を示したもの

```
** BEFORE RUN
----------------------------------------
NO 1        7D00        3E
NO 2        7D01        80
NO 3        7D02        21
NO 4        7D03        19
NO 5        7D04        7D
NO 6        7D05        ED
NO 7        7D06        4B
NO 8        7D07        17
NO 9        7D08        7D
NO 10       7D09        56
NO 11       7D0A        ED
NO 12       7D0B        A1
NO 13       7D0C        F2
NO 14       7D0D        10
NO 15       7D0E        7D
NO 16       7D0F        7A
NO 17       7D10        EA
NO 18       7D11        09
NO 19       7D12        7D
NO 20       7D13        32
NO 21       7D14        1E
NO 22       7D15        7D
NO 23       7D16        C9
NO 24       7D17        05
NO 25       7D18        00
NO 26       7D19        FF
NO 27       7D1A        FE
NO 28       7D1B        04
NO 29       7D1C        7F
NO 30       7D1D        20
NO 31       7D1E        00
----------------------------------------
```

```
** AFFTER RUN
----------------------------------------
NO 1        7D00        3E
NO 2        7D01        80
NO 3        7D02        21
NO 4        7D03        19
NO 5        7D04        7D
NO 6        7D05        ED
NO 7        7D06        4B
NO 8        7D07        17
NO 9        7D08        7D
NO 10       7D09        56
NO 11       7D0A        ED
NO 12       7D0B        A1
NO 13       7D0C        F2
NO 14       7D0D        10
NO 15       7D0E        7D
NO 16       7D0F        7A
NO 17       7D10        EA
NO 18       7D11        09
NO 19       7D12        7D
NO 20       7D13        32
NO 21       7D14        1E
NO 22       7D15        7D
NO 23       7D16        C9
NO 24       7D17        05
NO 25       7D18        00
NO 26       7D19        FF
NO 27       7D1A        FE
NO 28       7D1B        04
NO 29       7D1C        7F
NO 30       7D1D        20
NO 31       7D1E        7F
----------------------------------------
```

《第11－4図》 データを変えて，もう一度実験してみた場合

12 ある条件を満たすまで、同じ命令を繰り返すという命令グループ

Z80には，連続したメモリーに入っているデータを，ブロックごと移動したり，ブロックの中から特定のデータを選び出したりするときに非常に効率のよい命令があります。こういった命令は，すべて条件判断およびリピート命令を含んでいます。これらをここで説明するのですが，次のようなものがあります。

1．ロード命令系，LDIR，LDDR

2．比較命令系，CPIR，CPDR

3．入出力命令系，INIR，INDR，OTIR，OTDR

一つ一つ説明していくことにしましょう。

ロード命令系

ＬＤＩＲ，ＬＤＤＲの二つがありますが，それぞれを記号で表わすと次のようになります。

ＬＤＩＲ命令の場合

① Ｍ（ＤＥ）←Ｍ（ＨＬ）

② ＤＥ←ＤＥ＋１

③ ＨＬ←ＨＬ＋１

④ ＢＣ←ＢＣ－１

⑤ ＩＦ　ＢＣ≠０　GO TO　①

ＬＤＤＲ命令の場合

① Ｍ（ＤＥ）←Ｍ（ＨＬ）

② ＤＥ←ＤＥ－１

③ ＨＬ←ＨＬ－１

④ ＢＣ←ＢＣ－１

⑤ ＩＦ　ＢＣ≠０　GO TO　①

これからわかるように，レジスタペアＢＣに入れられたデータ数の分だけロード命令が繰り返されます。

似たような命令が二つありますが，なぜ二つも用意されているかおわかりでしょうか。

たとえば，７Ｄ００～７ＤＦＦのデータを，７Ｄ０２～７Ｅ０１へ移動したい，という場合には，ＬＤＩＲ命令は使えません。移動されたデータが移動前のデータを消してしまうからです。ここでは，ＨＬ＝７ＤＦＦ，ＢＣ＝７Ｅ０１として，ＬＤＤＲ命令を使います。

これらの命令が各フラグに与える影響は，次の通りです。

S　：　影響をうけない

Z　：　影響をうけない

P/V：　必ずゼロになる

CY：　影響をうけない

機械語との対応は次の通りです。

LDIR…………ED・B0

LDDR…………ED・B8

比較命令系

CPIR命令と，CPDR命令の二つがありますが，これらを，記号で表わすと次のようになります。

CPIR命令の場合

①IF A=M(HL) THEN Z=1, ELSE Z=0

②HL←HL+1

③BC←BC−1

④IF (BC≠0) Λ (Z=0) GO TO ①

CPDR命令の場合

①IF A=M(HL) THEN Z=1, ELSE Z=0

②HL←HL−1

③BC←BC−1

④IF (BC≠0) Λ (Z=0) GO TO ①

つまり，どちらも，レジスタAの内容のものと同じものがブロック内に見つかるか，または，ブロック内のデータをすべてチェックし終わるかするまでは，同じことを繰り返すわけです。

ブロック内に同じものがあったのか，そうではなく，すべてのデータをチェックし終わったのかの区別は，命令終了後のフラグの状態から判断します。この命令がフラグに与える影響を，次にまとめてみましょう。

S　：　Aと比較したメモリーの内容が，Aよりも大きかった場合（正負を考慮した上で）にS=1，そうでなかった場合，S=0

Z　：　Aと比較したメモリーの内容が，Aと同じであった場合，Z=1，そうでなかった場合，Z=0

P/V：　命令終了後BC=0ならば，P/V=0,そうでなければ，P/V=1

CY：　影響をうけない

なお，これらの命令の機械語との対応は次の通りです。

CPIR…………ED・B1

CPDR…………ED・B9

入出力命令系

入力系にINIR，INDR，出力系にOTIR，OTDRの命令があります。それぞれを記号で表わすと次のようになります。

INIR命令の場合

①　M(HL)←I/O (C)

②　HL←HL+1

③　B←B−1

INDR命令の場合

①　M(HL)←I/O (C)

②　HL←HL−1

③　B←B−1

OTIR命令の場合

①　I/O (C)←M(HL)

②　HL←HL+1

③　B←B−1

OTDRの場合

①　I/O (C)←M(HL)

②　HL←HL−1

③　B←B−1

レジスタBに，入出力したいデータの数を入れておけば，それらすべてが処理されるまで，この命令は続けられます。

これらの命令が各フラグに与える影響はみな同じで，次のようになります。

S　：　不定

Z　：　命令終了時点で　Z=1

P/V：　不定

CY：　影響をうけない

なお，これらの命令の機械語との対応は次の通りです。

INIR…………ED・B2

INDR…………ED・BA

OTIR…………ED・B3

OTDR…………ED・BB

LDDR命令を使ったテストプログラム

ＬＤＤＲ命令を説明した本文中で，ふれましたが，すでにメモリーに書かれている内容を，指定した分だけ後のメモリーに移すプログラムを作ってみました。考え方は，**第12―１図**に示した通りです。

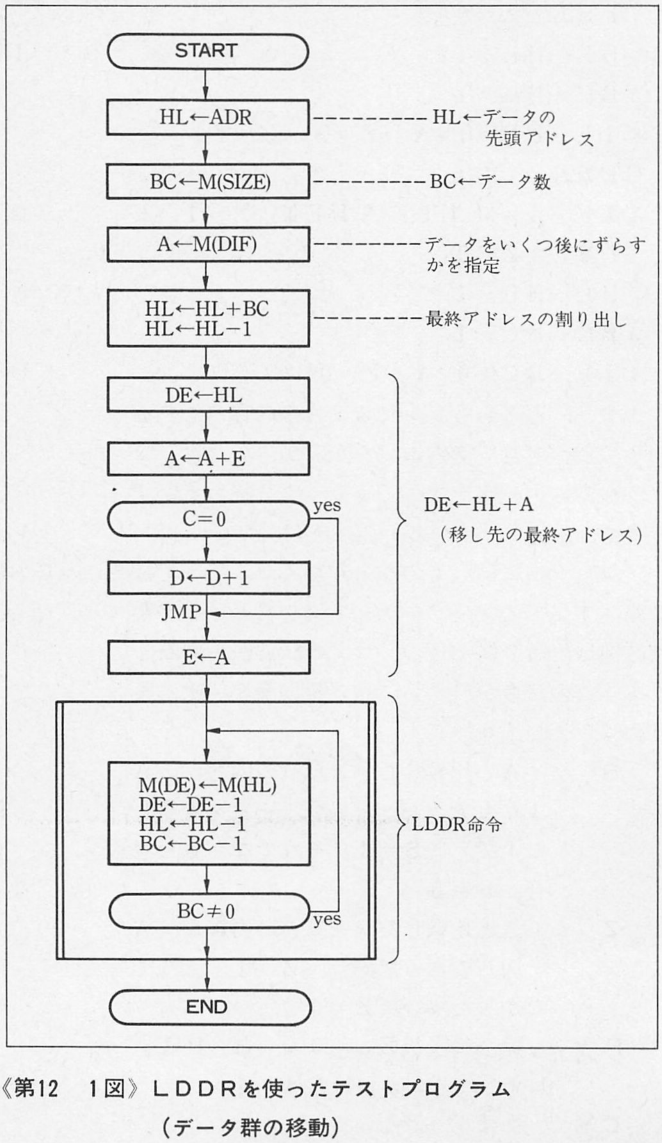
ＨＬに移したいデータの先頭アドレスを，レジスタペアＢＣに，移したいデータの数を，そしてレジスタＡに，いくらずらすかを指定する数を入れておきます。

ここから，移したいデータの最終アドレス，移し先の最終アドレスを計算して，それぞれＨＬ，ＤＥのレジスタペアに入れ，ここでＬＤＤＲ命令を実行する，というものです。

これをもとに，アセンブラプログラミングしたものが**第12―２図**です。

ここでは，データの数５，ずらす大きさを２，として，プログラムしてあります。

ここから得られた機械語を，直接メモリーに書き込み，命令実行前と，実行後のメモリーの内容を比較したものが**第12―３図**です。所定の働きをしていることが，ここから確認できます。



《第12　1図》ＬＤＤＲを使ったテストプログラム
（データ群の移動）

```
MACRO-80 3.3    09-MAY-79        PAGE    1
                                00100  ;        TEST PROGRAM ( LDDR )
                                00200  .Z80
0000'                           00300  ASEG
                                00400  ORG      7D00H
                                00500  ;
7D00    21 7D19                 00600           LD      HL,ADR          ; HL=ADR
7D03    ED 4B 7D16              00700           LD      BC,(SIZE)       ; BC=M(SIZE+1)M(SIZE)
7D07    3A 7D18                 00800           LD      A,(DIF)         ; A=M(DIF)
                                00900  ;
7D0A    09                      01000           ADD     HL,BC           ; HL=HL+BC
7D0B    2B                      01050           DEC     HL              ; HL=HL-1
7D0C    54                      01100           LD      D,H             ;
7D0D    5D                      01150           LD      E,L             ; DE=HL
7D0E    83                      01200           ADD     A,E             ; A=A+E
7D0F    30 01                   01300           JR      NC,JMP          ; IF CY=0 GOTO JMP
7D11    14                      01400           INC     D               ; D=D+1
7D12    5F                      01500  JMP:     LD      E,A             ; E=A  ( DE=DE+A )
                                01600  ;
7D13    ED B8                   01700           LDDR
7D15    C9                      01800           RET                     ; RETURN
                                01900  ;
7D16    05                      02000  SIZE:    DB      5               ; M(SIZE)=05
7D17    00                      02100           DB      0               ; M(SIZE+1)=00
7D18    02                      02200  DIF:     DB      2               ; M(DIF)=02
7D19    01                      02300  ADR:     DB      1               ; M(ADR)=1
7D1A    02                      02400           DB      2               ; M(ADR+1)=2
7D1B    03                      02500           DB      3               ; M(ADR+2)=3
7D1C    04                      02600           DB      4               ; M(ADR+3)=4
7D1D    05                      02700           DB      5               ; M(ADR+4)=5
7D1E    00                      02800           DB      0               ; M(ADR+5)=0
7D1F    00                      02900           DB      0               ; M(ADR+6)=0
7D20    00                      03000           DB      0               ; M(ADR+7)=0
                                03100           END
```

《第12―2図》第12―1図をもとに，アセンブラプログラミングしたもの

** BEFORE RUN

NO 1	7D00	21
NO 2	7D01	19
NO 3	7D02	7D
NO 4	7D03	ED
NO 5	7D04	4B
NO 6	7D05	16
NO 7	7D06	7D
NO 8	7D07	3A
NO 9	7D08	18
NO 10	7D09	7D
NO 11	7D0A	09
NO 12	7D0B	2B
NO 13	7D0C	54
NO 14	7D0D	5D
NO 15	7D0E	83
NO 16	7D0F	30
NO 17	7D10	01
NO 18	7D11	14
NO 19	7D12	5F
NO 20	7D13	ED
NO 21	7D14	B8
NO 22	7D15	C9
NO 23	7D16	05
NO 24	7D17	00
NO 25	7D18	02
NO 26	7D19	01
NO 27	7D1A	02
NO 28	7D1B	03
NO 29	7D1C	04
NO 30	7D1D	05
NO 31	7D1E	00
NO 32	7D1F	00
NO 33	7D20	00

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	19
NO 3	7D02	7D
NO 4	7D03	ED
NO 5	7D04	4B
NO 6	7D05	16
NO 7	7D06	7D
NO 8	7D07	3A
NO 9	7D08	18
NO 10	7D09	7D
NO 11	7D0A	09
NO 12	7D0B	2B
NO 13	7D0C	54
NO 14	7D0D	5D
NO 15	7D0E	83
NO 16	7D0F	30
NO 17	7D10	01
NO 18	7D11	14
NO 19	7D12	5F
NO 20	7D13	ED
NO 21	7D14	B8
NO 22	7D15	C9
NO 23	7D16	05
NO 24	7D17	00
NO 25	7D18	02
NO 26	7D19	01
NO 27	7D1A	02
NO 28	7D1B	01
NO 29	7D1C	02
NO 30	7D1D	03
NO 31	7D1E	04
NO 32	7D1F	05
NO 33	7D20	00

《第12—3図》第12—2図の機械語を直接メモリーに入れ，実行前と実行後のそれを比較したもの

13 Z80用 基本プログラム

Z80用のソフトウエアで，最も基本的なものをここに紹介してみましょう。それは

1．データを大きい順に並べかえるプログラム
2．8ビット×8ビットの掛け算のプログラム
3．16ビット×16ビットの掛け算のプログラム
4．16ビット÷8ビットの割り算のプログラム
5．16ビットの平方根を求めるプログラム

の五つですが，これらは，サブルーチンとして活用することもできるよう考慮されています。

一つ一つ簡単に説明していきましょう。

データを大きい順に並べかえるプログラム

代表的なアルゴリズムが二つあります。このアルゴリズムについて詳しいことを，お知りになりたい方は，雑誌「マイコン2月号」（1979年）を参考にしてください。ここでは**第13—1図**のフローチャートの示すアルゴリズムを用いて話しをすすめていきます。

このプログラムがサブルーチンとして用いられることを考慮して，スタック領域を使わないように工夫がしてあります。

レジスタペアHLに，データの先頭アドレスを入れ，レジスタBにデータ数を入れてから，LOOP1というシンボルのつけられたところから実行すれば，データを大きい順に並べかえられます。

第13—2図は，**第13—1図**をもとにアセンブラプログラミングしたものです。

ここでは，データ数は5，データは，3，7，－2，3，10（すべて10進数）としてあります。

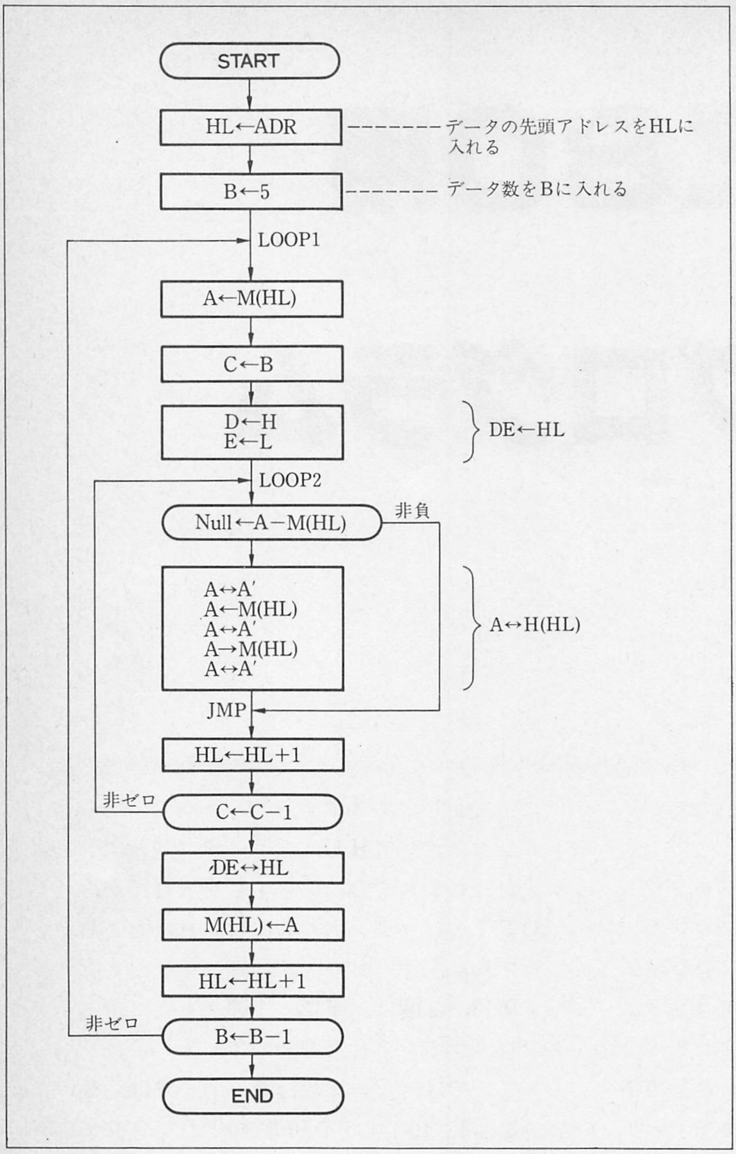
ここから与えられた機械語や，データを直接メモリーに入れ，プログラム実行前と実行後のメモリーの内容を比較したものが**第13—3図**です。

データが，0A，07，03，03，　FEつまり，10，7，3，3，－2の順になっていることが確認できます。

なお，ここでは8ビットが，－128～＋127　を表現しているものとしてプログラムを作ってありますが，8ビットが，0～255 を表現しているとみなす場合は，**第13—2図**の行番号1400の命令を

1400　JR　C，JMP

とかえればよいでしょう。この場合は，オーバーフローに対する心配はなくなります。また，全体



《第13—1図》

データを大きい順に並べかえるプログラム（オーバーフローのある場合は正しい結果は得られない）

として，1ワード分，プログラムが短くなります。

8ビット×8ビットの掛け算のプログラム

ここでは，H×Eの結果がレジスタペアHLに入るものを考えてみました。

アルゴリズムについては，「わかるアセンブラプログラミング」（電波新聞社）の第1章を参考にしてください。

フローチャートを**第13—4図**に示しました。HLを左へシフトさせるのを　HL←HL+HL　で行なっている点　HL←HL+E　を　D=0として，　HL←HL+DE　としている点がポイントです。

このフローチャートをもとにして，アセンブラプログラミングしたものが，**第13—5図**です。ここでは，メモリーM(DATA1)に入っている内容と，M(DATA2)に入っている内容を計算して，結果をM(ANS+1)とM(ANS)に入れるようになっています。例として，M(DATA1)=FF，M(DATA2)=FFとしてあります。

ここから得られた機械語を直接メモリーに入れ，実行前と実行後のメモリーの内容を示したものが，

```
MACRO-80 3.3    09-MAY-79       PAGE    1
                          00100  ;        TEST PROGRAM ( SORTING DATA )
                          00200  .Z80
0000'                     00300  ASEG
                          00400  ORG      7D00H
                          00500  ;
7D00    21 7D1C           00600           LD      HL,ADR          ; HL=ADR
7D03    06 05             00700           LD      B,5             ; B=5
7D05    7E                00800  LOOP1:   LD      A,(HL)          ; A=M(HL)
7D06    48                00900           LD      C,B             ; C=B
7D07    54                01000           LD      D,H             ;
7D08    5D                01100           LD      E,L             ; DE=HL
7D09    BE                01300  LOOP2:   CP      (HL)            ; NULL=A-M(HL)
7D0A    F2 7D12           01400           JP      P,JMP           ; IF S=0 GOTO JMP
7D0D    08                01500           EX      AF,AF'          ; A<->A'
7D0E    7E                01510           LD      A,(HL)          ; A=M(HL)
7D0F    08                01520           EX      AF,AF'          ; A<->A'
7D10    77                01530           LD      (HL),A          ; M(HL)=A
7D11    08                01540           EX      AF,AF'          ; A<->A'
7D12    23                01600  JMP:     INC     HL              ; HL=HL+1
7D13    0D                01700           DEC     C               ; C=C-1
7D14    20 F3             01800           JR      NZ,LOOP2        ; IF C<>0 GOTO LOOP2
7D16    EB                01900           EX      DE,HL           ; DE<->HL
7D17    77                02000           LD      (HL),A          ; M(HL)=A
7D18    23                02100           INC     HL              ; HL=HL+1
7D19    10 EA             02200           DJNZ    LOOP1           ; B=B-1 : IF B<>0 GOTO LOOP1
7D1B    C9                02300           RET                     ; RETURN
                          02400  ;
7D1C    03                02600  ADR:     DB      3               ; DATA1=3
7D1D    07                02700           DB      7               ; DATA2=7
7D1E    FE                02800           DB      -2D             ; DATA3=-2
7D1F    03                02900           DB      3               ; DATA4=3
7D20    0A                03000           DB      10D             ; DATA5=10
                          03100  ;
                          03200           END
```

《**第13—2図**》**第13—1図をもとに，アセンブラプログラミングしたもの**

第13—6図です。ここで，実行後，M（ANS＋1）＝FE，M（ANS）＝01になっていることが確認できます。

FF×FF＝FE01が計算できているわけです。

16ビット×16ビットの掛け算のプログラム

考え方は，8ビット×8ビットの場合と同じです。ただ，32ビットどうしの足し算が一度にできないので，DE↔HL　命令をうまく使って，**第13—7図**のフローチャートのようにします。この部分がこのプログラムのポイントでしょう。

これをもとにアセンブラプログラミングしたものが**第13—8図**です。

ここでは，メモリーM(DATA1+1) M(DATA1）に入っている内容と，M（DATA2＋1）M（DATA2）に入っている内容を掛けて，その結果を，M（ANS＋3）～M（ANS）に入れるようになっています。

また，例として，FFFF×FFFFを計算するようにしてあります。

これを実際に確かめるために，ここから得られた機械語を直接メモリーに書き込んで，実行前後の変化をみようとしたのが，**第13—9図**です。実行後

M（7D27)＝01

M（7D28)＝00

M（7D29)＝FE

M（7D2A)＝FF

** BEFORE RUN

NO 1	7D00	21
NO 2	7D01	1C
NO 3	7D02	7D
NO 4	7D03	06
NO 5	7D04	05
NO 6	7D05	7E
NO 7	7D06	48
NO 8	7D07	54
NO 9	7D08	5D
NO 10	7D09	BE
NO 11	7D0A	F2
NO 12	7D0B	12
NO 13	7D0C	7D
NO 14	7D0D	08
NO 15	7D0E	7E
NO 16	7D0F	08
NO 17	7D10	77
NO 18	7D11	08
NO 19	7D12	23
NO 20	7D13	0D
NO 21	7D14	20
NO 22	7D15	F3
NO 23	7D16	EB
NO 24	7D17	77
NO 25	7D18	23
NO 26	7D19	10
NO 27	7D1A	EA
NO 28	7D1B	C9
NO 29	7D1C	03
NO 30	7D1D	07
NO 31	7D1E	FE
NO 32	7D1F	03
NO 33	7D20	0A

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	1C
NO 3	7D02	7D
NO 4	7D03	06
NO 5	7D04	05
NO 6	7D05	7E
NO 7	7D06	48
NO 8	7D07	54
NO 9	7D08	5D
NO 10	7D09	BE
NO 11	7D0A	F2
NO 12	7D0B	12
NO 13	7D0C	7D
NO 14	7D0D	08
NO 15	7D0E	7E
NO 16	7D0F	08
NO 17	7D10	77
NO 18	7D11	08
NO 19	7D12	23
NO 20	7D13	0D
NO 21	7D14	20
NO 22	7D15	F3
NO 23	7D16	EB
NO 24	7D17	77
NO 25	7D18	23
NO 26	7D19	10
NO 27	7D1A	EA
NO 28	7D1B	C9
NO 29	7D1C	0A
NO 30	7D1D	07
NO 31	7D1E	03
NO 32	7D1F	03
NO 33	7D20	FE

《第13―3図》第13―2図の機械語を試したもの

となっていることが確認できます。つまり，

ＦＦＦＦ×ＦＦＦＦ＝ＦＦＦＥ０００１

が正確に計算されているわけです。

16ビット÷8ビットの割り算のプログラム

8ビット×8ビットの場合

ＨＬ←Ｈ×Ｌ

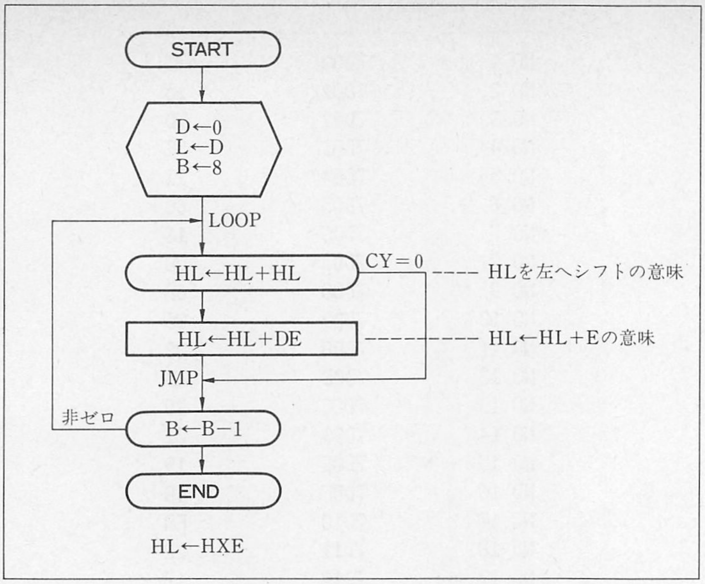
としたので，16ビット÷8ビットは

Ｈ←―ＨＬ÷Ｅ
Ｌ←―――┘

としました。つまり，レジスタペアＨＬに入っている16ビット以下の数を，レジスタＥに入っている8ビットの数で割り，そのときの商をレジスタＨに，余りをＬに入れるというものです。

ただし，ゼロで割ろうとしたり，商が8ビットを越えてしまう場合について，警告を発するようにはしてありません。もし，この割り算のプログラムを，システム中のサブルーチンとして用いる場合には，そのシステムに合った警告を発する工夫をサブルーチン内，あるいはサブルーチンへ移る直前のメイン側に施さなければなりません。

サブルーチン内で行なう場合は，メインから移



《第13—4図》ＨＬ←Ｈ×Ｅのアルゴリズム

```
MACRO-80 3.3    09-MAY-79       PAGE    1
                        ;       TEST PROGRAM ( 8 BIT MULTIPLICATION )
                        .Z80
0000                    ASEG
                        ORG     7D00H
                        ;
7D00    21 7D15                 LD      HL,DATA1        ; HL=DATA1
7D03    5E                      LD      E,(HL)          ; E=M(DATA1)
7D04    23                      INC     HL              ; HL=HL+1
7D05    66                      LD      H,(HL)          ; H=M(DATA2)
                        ;
7D06    16 00                   LD      D,0             ; D=0
7D08    6A                      LD      L,D             ; L=0
7D09    06 08                   LD      B,8             ; B=8
7D0B    29              LOOP:   ADD     HL,HL           ; HL=2*HL
7D0C    30 01                   JR      NC,JMP          ; IF CY=0 GOTO JMP
7D0E    19                      ADD     HL,DE           ; HL=HL+E
7D0F    10 FA           JMP:    DJNZ    LOOP            ; B=B-1 : IF B<>0 GOTO LOOP
7D11    22 7D17                 LD      (ANS),HL        ; M(ANS+1)M(ANS)=HL
7D14    C9                      RET                     ; RETURN
                        ;
7D15    FF              DATA1:  DB      0FFH            ; M(DATA1)=FF
7D16    FF              DATA2:  DB      0FFH            ; M(DATA2)=FF
7D17    00              ANS:    DB      0               ; M(ANS)=0
7D18    00                      DB      0               ; M(ANS+1)=0
                                END
```

《第13—5図》第13—4図をもとに，ＨＬ＝Ｈ×Ｅをアセンブラプログラミングしたもの

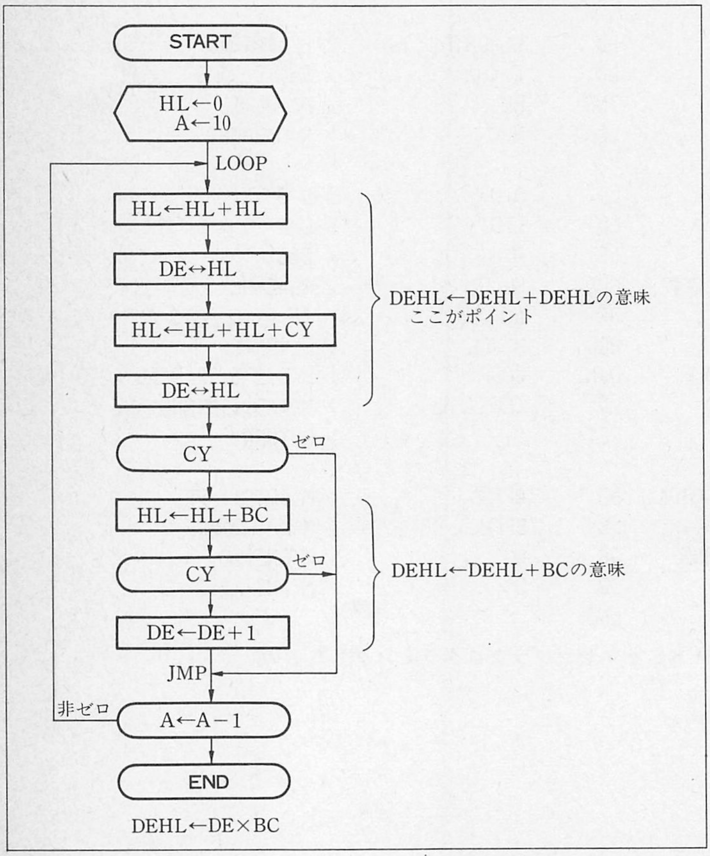
** BEFORE RUN

NO 1	7D00	21
NO 2	7D01	15
NO 3	7D02	7D
NO 4	7D03	5E
NO 5	7D04	23
NO 6	7D05	66
NO 7	7D06	16
NO 8	7D07	00
NO 9	7D08	6A
NO 10	7D09	06
NO 11	7D0A	08
NO 12	7D0B	29
NO 13	7D0C	30
NO 14	7D0D	01
NO 15	7D0E	19
NO 16	7D0F	10
NO 17	7D10	FA
NO 18	7D11	22
NO 19	7D12	17
NO 20	7D13	7D
NO 21	7D14	C9
NO 22	7D15	FF
NO 23	7D16	FF
NO 24	7D17	00
NO 25	7D18	00

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	15
NO 3	7D02	7D
NO 4	7D03	5E
NO 5	7D04	23
NO 6	7D05	66
NO 7	7D06	16
NO 8	7D07	00
NO 9	7D08	6A
NO 10	7D09	06
NO 11	7D0A	08
NO 12	7D0B	29
NO 13	7D0C	30
NO 14	7D0D	01
NO 15	7D0E	19
NO 16	7D0F	10
NO 17	7D10	FA
NO 18	7D11	22
NO 19	7D12	17
NO 20	7D13	7D
NO 21	7D14	C9
NO 22	7D15	FF
NO 23	7D16	FF
NO 24	7D17	01
NO 25	7D18	FE

《第13－6図》第13－5図の機械語を試したもの



《第13－7図》DEHL←DE×BCのアルゴリズム

```
    MACRO-80 3.3    09-MAY-79       PAGE    1
                            ;       TEST PROGRAM ( 16 BIT MULTIPLICATION )
                            .Z80
0000'                       ASEG
                            ORG     7D00H
                            ;
7D00    ED 5B 7D23                  LD      DE,(DATA1)      ; DE=M(DATA1+1)M(DATA1)
7D04    ED 4B 7D25                  LD      BC,(DATA2)      ; BC=M(DATA2+1)M(DATA2)
                            ;
7D08    21 0000                     LD      HL,0            ; HL=0
7D0B    3E 10                       LD      A,10H           ; A=10
7D0D    29                  LOOP:   ADD     HL,HL           ; HL=2*HL
7D0E    EB                          EX      DE,HL           ;
7D0F    ED 6A                       ADC     HL,HL           ; HL=2*HL+CY
7D11    EB                          EX      DE,HL           ; ( DEHL=2*DEHL )
7D12    30 04                       JR      NC,JMP          ; IF CY=0 GOTO JMP
7D14    09                          ADD     HL,BC           ; HL=HL+BC
7D15    30 01                       JR      NC,JMP          ; IF CY=0 GOTO JMP
7D17    13                          INC     DE              ; DE=DE+1
7D18    3D                  JMP:    DEC     A               ; A=A-1
7D19    20 F2                       JR      NZ,LOOP         ; IF A<>0 GOTO LOOP
                            ;
7D1B    22 7D27                     LD      (ANS),HL        ;
7D1E    ED 53 7D29                  LD      (ANS+2),DE      ; M(ANS+3)M(ANS+2)M(ANS+1)M(ANS)=DEHL
7D22    C9                          RET                     ; RETURN
                            ;
7D23    FF                  DATA1:  DB      0FFH            ;
7D24    FF                          DB      0FFH            ; M(DATA1+1)M(DATA1)=FFFF
7D25    FF                  DATA2:  DB      0FFH            ;
7D26    FF                          DB      0FFH            ; M(DATA2+1)M(DATA2)=FFFF
7D27    00                  ANS:    DB      0               ;
7D28    00                          DB      0               ;
7D29    00                          DB      0               ;
7D2A    00                          DB      0               ; M(ANS+3)M(ANS+2)M(ANS+1)M(ANS)=0
                                    END
```

《第13－8図》第13－7図をもとに，アセンブラプログラミングしたもの

った直後に，HとEを比較して，HがE以上である場合（もちろん第7ビットが1である場合も正とみなして）に，警告を発するようにするか，警告発生用のサブルーチンを別に作っておいて，そこにジャンプさせるようにすればよいでしょう。

サブルーチンを簡素にしたいのであれば，HがE以上であったら，CYが1になるような比較のし方をして，サブルーチンを終わらせるようにする方法もあります。ただしこの場合，普通に実行できた場合には，CYがゼロで終わるようにしておかなければ意味ありません。しかし，こうしておくと，このサブルーチンは色々なシステムで使えることになります（つまり，システムによって警告の出し方がちがう場合でも）。

アルゴリズムについて詳しいことをお知りになりたい方は，「わかるアセンブラプログラミング」（電波新聞社）の第一章を参考にしてください。

フローチャートを**第13—10図**に書いておきました。H，L，Eのレジスタ以外に，AとBのレジスタを使っています。

これをもとに，アセンブラプログラミングしたものが，**第13—11図**です。

ここでは，M(DATA１＋１)M(DATA１）に入っている内容をM（DATA２）で割って，商をM（ANS＋１)，余りをM（ANS）に入れるようにしてあります。また例として，ＦＥ０１÷ＦＦを計算するようにしてあります。

これを実際にためしてみたものが**第13—12図**です。

実行後，

M（７D25)＝ＦＦ………商

M（７日24)＝００………余り

となっていることが確認できます。

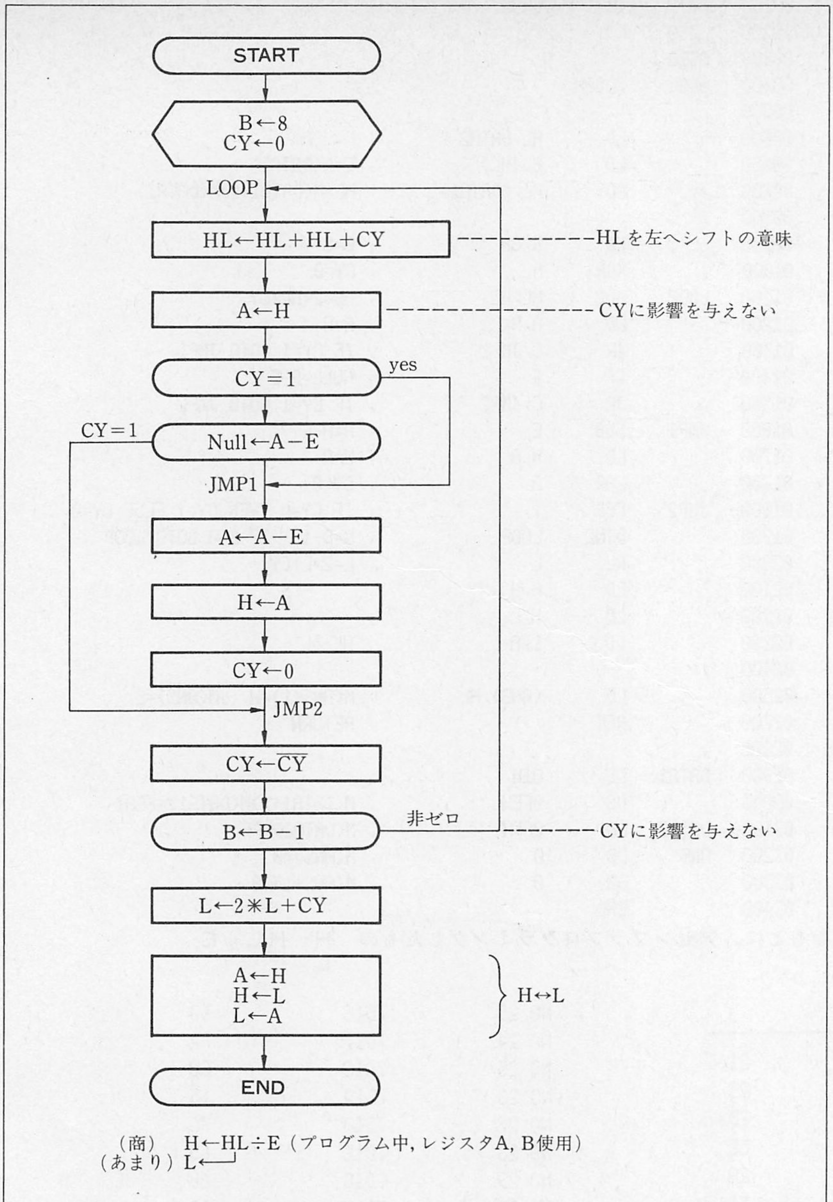
** BEFORE RUN

NO 1	7D00	ED
NO 2	7D01	5B
NO 3	7D02	23
NO 4	7D03	7D
NO 5	7D04	ED
NO 6	7D05	4B
NO 7	7D06	25
NO 8	7D07	7D
NO 9	7D08	21
NO 10	7D09	00
NO 11	7D0A	00
NO 12	7D0B	3E
NO 13	7D0C	10
NO 14	7D0D	29
NO 15	7D0E	EB
NO 16	7D0F	ED
NO 17	7D10	6A
NO 18	7D11	EB
NO 19	7D12	38
NO 20	7D13	04
NO 21	7D14	09
NO 22	7D15	30
NO 23	7D16	01
NO 24	7D17	13
NO 25	7D18	3D
NO 26	7D19	20
NO 27	7D1A	F2
NO 28	7D1B	22
NO 29	7D1C	27
NO 30	7D1D	7D
NO 31	7D1E	ED
NO 32	7D1F	53
NO 33	7D20	29
NO 34	7D21	7D
NO 35	7D22	C9
NO 36	7D23	FF
NO 37	7D24	FF
NO 38	7D25	FF
NO 39	7D26	FF
NO 40	7D27	00
NO 41	7D28	00
NO 42	7D29	00
NO 43	7D2A	00

《第13－9図》第13－8図の機械語を試したもの

** AFFTER RUN

NO 1	7D00	ED
NO 2	7D01	5B
NO 3	7D02	23
NO 4	7D03	7D
NO 5	7D04	ED
NO 6	7D05	4B
NO 7	7D06	25
NO 8	7D07	7D
NO 9	7D08	21
NO 10	7D09	00
NO 11	7D0A	00
NO 12	7D0B	3E
NO 13	7D0C	10
NO 14	7D0D	29
NO 15	7D0E	EB
NO 16	7D0F	ED
NO 17	7D10	6A
NO 18	7D11	EB
NO 19	7D12	38
NO 20	7D13	04
NO 21	7D14	09
NO 22	7D15	30
NO 23	7D16	01
NO 24	7D17	13
NO 25	7D18	3D
NO 26	7D19	20
NO 27	7D1A	F2
NO 28	7D1B	22
NO 29	7D1C	27
NO 30	7D1D	7D
NO 31	7D1E	ED
NO 32	7D1F	53
NO 33	7D20	29
NO 34	7D21	7D
NO 35	7D22	C9
NO 36	7D23	FF
NO 37	7D24	FF
NO 38	7D25	FF
NO 39	7D26	FF
NO 40	7D27	01
NO 41	7D28	00
NO 42	7D29	FE
NO 43	7D2A	FF



《第13—10図》ＨＬ÷Ｅの商をＨ，余りをＬ，に入れるアルゴリズム

16ビットの平方根を求めるプログラム

小数点以下を四捨五入する場合，16ビットの平方根は，最大で9ビットになります。したがってここでは，

$$BC \leftarrow \sqrt{HL}$$

というプログラムを考えてみました。答が9ビットになる場合，最上位のビットをＣＹを用いて表わす方法もありますが，このプログラムをそのように直すのは，わけないことです。

フローチャートを**第13—14図**に書きましたが，こんなもので，16ビットの平方根が，しかも小数点以下を四捨五入した場合の正確な値が求まるのか，と思われる方もいると思いますが，確かにできます。誤差はありません（小数点以下を四捨五入した場合の整数の範囲で）。またルートゼロはゼロと答えてくれるので，全く警告発生について考える必要もありません。

```
  MACRO-80 3.3    09-MAY-79        PAGE    1
                            00100  ;          TEST PROGRAM ( DIVISION )
                            00200  .Z80
0000'                       00300  ASEG
                            00400  ORG        7D00H
                            00500  ;
7D00    21 7D23             00600             LD     HL,DATA2       ;
7D03    5E                  00650             LD     E,(HL)         ; E=M(DATA2)
7D04    2A 7D21             00700             LD     HL,(DATA1)     ; HL=M(DATA1+1)M(DATA1)
                            00800  ;
7D07    06 08               00900             LD     B,8H           ; B=8
7D09    AF                  01000             XOR    A              ; CY=0
7D0A    ED 6A               01100  LOOP:      ADC    HL,HL          ; HL=2*HL+CY
7D0C    7C                  01200             LD     A,H            ; A=H
7D0D    38 03               01300             JR     C,JMP1         ; IF CY=1 GOTO JMP1
7D0F    BB                  01400             CP     E              ; NULL=A-E
7D10    38 03               01500             JR     C,JMP2         ; IF CY=1 GOTO JMP2
7D12    93                  01600  JMP1:      SUB    E              ; A=A-E
7D13    67                  01700             LD     H,A            ; H=A
7D14    AF                  01750             XOR    A              ; CY=0
7D15    3F                  01800  JMP2:      CCF                   ; IF CY=0 THEN CY=1 ELSE CY=0
7D16    10 F2               01900             DJNZ   LOOP           ; B=B-1 : IF B=0 GOTO LOOP
7D18    CB 15               02000             RL     L              ; L=2*L+CY
7D1A    7C                  02100             LD     A,H            ;
7D1B    65                  02200             LD     H,L            ;
7D1C    6F                  02300             LD     L,A            ; H<->L
                            02400  ;
7D1D    22 7D24             02500             LD     (ANS),HL       ; M(ANS+1)=H : M(ANS)=L
7D20    C9                  02700             RET                   ; RETURN
                            02800  ;
7D21    01                  02900  DATA1:     DB     01H            ;
7D22    FE                  03000             DB     0FEH           ; M(DATA1+1)M(DATA1)=FE01
7D23    FF                  03100  DATA2:     DB     0FFH           ; M(DATA2)=FF
7D24    00                  03200  ANS:       DB     0              ; M(ANS)=0
7D25    00                  03300             DB     0              ; M(ANS+1)=0
                            03400             END
```

《第13—11図》第13—10図をもとに，アセンブラプログラミングしたもの　H←HL÷E
L ←

```
** BEFORE RUN
--------------------------------------
NO 1        7D00        21
NO 2        7D01        23
NO 3        7D02        7D
NO 4        7D03        5E
NO 5        7D04        2A
NO 6        7D05        21
NO 7        7D06        7D
NO 8        7D07        06
NO 9        7D08        08
NO 10       7D09        AF
NO 11       7D0A        ED
NO 12       7D0B        6A
NO 13       7D0C        7C
NO 14       7D0D        38
NO 15       7D0E        03
NO 16       7D0F        BB
NO 17       7D10        38
NO 18       7D11        03
NO 19       7D12        93
NO 20       7D13        67
NO 21       7D14        AF
NO 22       7D15        3F
NO 23       7D16        10
NO 24       7D17        F2
NO 25       7D18        CB
NO 26       7D19        15
NO 27       7D1A        7C
NO 28       7D1B        65
NO 29       7D1C        6F
NO 30       7D1D        22
NO 31       7D1E        24
NO 32       7D1F        7D
NO 33       7D20        C9
NO 34       7D21        01
NO 35       7D22        FE
NO 36       7D23        FF
NO 37       7D24        00
NO 38       7D25        00
--------------------------------------
```

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	23
NO 3	7D02	7D
NO 4	7D03	5E
NO 5	7D04	2A
NO 6	7D05	21
NO 7	7D06	7D
NO 8	7D07	06
NO 9	7D08	08
NO 10	7D09	AF
NO 11	7D0A	ED
NU 12	7D0B	6A
NO 13	7D0C	7C
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	BB
NO 17	7D10	38
NO 18	7D11	03
NO 19	7D12	93
NO 20	7D13	67
NO 21	7D14	AF
NO 22	7D15	3F
NO 23	7D16	10
NO 24	7D17	F2
NO 25	7D18	CB
NO 26	7D19	15
NO 27	7D1A	7C
NO 28	7D1B	65
NO 29	7D1C	6F
NO 30	7D1D	22
NO 31	7D1E	24
NO 32	7D1F	7D
NO 33	7D20	C9
NO 34	7D21	01
NO 35	7D22	FE
NO 36	7D23	FF
NO 37	7D24	00
NO 38	7D25	FF

《第13−12図》第13−11図の機械語を試してみたもの

** BEFORE RUN

NO 1	7D00	21
NO 2	7D01	23
NO 3	7D02	7D
NO 4	7D03	5E
NO 5	7D04	2A
NO 6	7D05	21
NO 7	7D06	7D
NO 8	7D07	06
NO 9	7D08	08
NO 10	7D09	AF
NO 11	7D0A	ED
NO 12	7D0B	6A
NO 13	7D0C	7C
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	BB
NO 17	7D10	38
NO 18	7D11	03
NO 19	7D12	93
NO 20	7D13	67
NO 21	7D14	AF
NO 22	7D15	3F
NO 23	7D16	10
NO 24	7D17	F2
NO 25	7D18	CB
NO 26	7D19	15
NO 27	7D1A	7C
NO 28	7D1B	65
NO 29	7D1C	6F
NO 30	7D1D	22
NO 31	7D1E	24
NO 32	7D1F	7D
NO 33	7D20	C9
NO 34	7D21	0A
NO 35	7D22	00
NO 36	7D23	03
NO 37	7D24	00
NO 38	7D25	00

** AFFTER RUN

NO 1	7D00	21
NO 2	7D01	23
NO 3	7D02	7D
NO 4	7D03	5E
NO 5	7D04	2A
NO 6	7D05	21
NO 7	7D06	7D
NO 8	7D07	06
NO 9	7D08	08
NO 10	7D09	AF
NO 11	7D0A	ED
NO 12	7D0B	6A
NO 13	7D0C	7C
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	BB
NO 17	7D10	38
NO 18	7D11	03
NO 19	7D12	93
NO 20	7D13	67
NO 21	7D14	AF
NO 22	7D15	3F
NO 23	7D16	10
NO 24	7D17	F2
NO 25	7D18	CB
NO 26	7D19	15
NO 27	7D1A	7C
NO 28	7D1B	65
NO 29	7D1C	6F
NO 30	7D1D	22
NO 31	7D1E	24
NO 32	7D1F	7D
NO 33	7D20	C9
NO 34	7D21	0A
NO 35	7D22	00
NO 36	7D23	03
NO 37	7D24	01
NO 38	7D25	03

《第13−13図》第13−11図のプログラムのデータを変えてみて実験したもの

これをもとに，アセンブラプログラミングしたものが，**第13—15図**です。

ここでは，M(DATA＋1)M(DATA) の平方根を求め，M(ANS＋1)M(ANS)に入れるようになっています。計算例として，ルートＦＦＦＦをするようになっています。

これを確かめてみたのが，**第13—16図**です。実行前と実行後のメモリーの内容が比較できます。

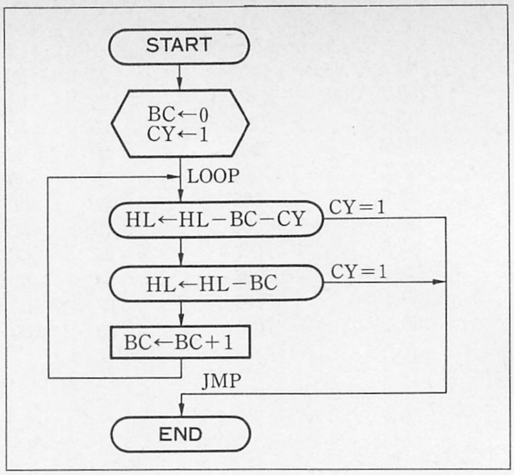
実行後に

M（７Ｄ１Ａ）＝０１

M（７Ｄ１９）＝００

となっているのが確信できます。つまり

$\sqrt{\text{FFFF}}$（$\sqrt{65535}$）＝１００（256）



《第13—14図》ＢＣ←$\sqrt{\text{HL}}$（小数点以下四捨五入）のアルゴリズム

```
      MACRO-80 3.3    09-MAY-79        PAGE    1
                                00100  ;        TEST PROGRAM ( BC=SQR(HL) )
                                00200  .Z80
0000'                           00300  ASEG
                                00400  ORG      7D00H
                                00500  ;
7D00    2A 7D17                 00600           LD      HL,(DATA)       ; HL=M(DATA)
                                00700  ;
7D03    01 0000                 00800           LD      BC,0            ; BC=0
7D06    37                      01000           SCF                     ; CY=1
7D07    ED 42                   01200  LOOP:    SBC     HL,BC           ; HL=HL-BC-CY
7D09    38 07                   01300           JR      C,JMP           ; IF CY=1 GOTO JMP
7D0B    ED 42                   01400           SBC     HL,BC           ; HL=HL-BC
7D0D    38 03                   01500           JR      C,JMP           ; IF CY=1 GOTO JMP
7D0F    03                      01600           INC     BC              ; BC=BC+1
7D10    18 F5                   01700           JR      LOOP            ; GOTO LOOP
7D12    ED 43 7D19              01800  JMP:     LD      (ANS),BC        ; M(ANS+1)M(ANS)=BC
7D16    C9                      01900           RET                     ; RETURN
                                02000  ;
7D17    FF                      02100  DATA:    DB      0FFH            ;
7D18    FF                      02200           DB      0FFH            ; M(DATA+1)M(DATA)=FFFF
7D19    00                      02300  ANS:     DB      0               ;
7D1A    00                      02400           DB      0               ; M(ANS+1)M(ANS)=0
                                02500           END
```

《第13—15図》ＢＣ←$\sqrt{\text{HL}}$（小数点以下四捨五入）のプログラム　ただし，ＨＬ＝ＦＦＦＦを例としてある

（小数点以下四捨五入）

を計算しているわけです。

第13—17図に　$\sqrt{0}$　の場合，**第13—18図**に$\sqrt{7}$，**第13—19図**に$\sqrt{6}$の場合をあげておきました。

$\sqrt{7}$≒2.646

$\sqrt{6}$≒2.449

ですから，小数点以下を四捨五入すると，一方は3に，一方は2になるはずです。ニュートン法などでは，このように小さい$\sqrt{7}$や$\sqrt{6}$の区別がなかなかつけられないのですが，ここでは，3と2に区別しているのがおわかりでしょう。

** BEFORE RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	FF
NO 25	7D18	FF
NO 26	7D19	00
NO 27	7D1A	00

** AFFTER RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	FF
NO 25	7D18	FF
NO 26	7D19	00
NO 27	7D1A	01

《第13—16図》第13—15図の機械語を試してみたもの　$\sqrt{\mathrm{FFFF}}=0／00$ の計算

** BEFORE RUN

** BEFORE RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	00
NO 25	7D18	00
NO 26	7D19	00
NO 27	7D1A	01

** AFFTER RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	00
NO 25	7D18	00
NO 26	7D19	00
NO 27	7D1A	00

《第13—17図》$\sqrt{0000}=0000$ を計算

** BEFORE RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	07
NO 25	7D18	00
NO 26	7D19	00
NO 27	7D1A	00

** AFFTER RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	07
NO 25	7D18	00
NO 26	7D19	03
NO 27	7D1A	00

《第13—18図》$\sqrt{0\ 0\ 0\ 7}$ = 0 0 0 3 を計算

** BEFORE RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	06
NO 25	7D18	00
NO 26	7D19	03
NO 27	7D1A	00

** AFFTER RUN

NO 1	7D00	2A
NO 2	7D01	17
NO 3	7D02	7D
NO 4	7D03	01
NO 5	7D04	00
NO 6	7D05	00
NO 7	7D06	37
NO 8	7D07	ED
NO 9	7D08	42
NO 10	7D09	38
NO 11	7D0A	07
NO 12	7D0B	ED
NO 13	7D0C	42
NO 14	7D0D	38
NO 15	7D0E	03
NO 16	7D0F	03
NO 17	7D10	18
NO 18	7D11	F5
NO 19	7D12	ED
NO 20	7D13	43
NO 21	7D14	19
NO 22	7D15	7D
NO 23	7D16	C9
NO 24	7D17	06
NO 25	7D18	00
NO 26	7D19	02
NO 27	7D1A	00

《第13—19図》$\sqrt{0\ 0\ 0\ 6}$ = 0 0 0 2 を計算

14 Z-80 実用プログラム制作のノウハウ

一つの簡単な完成したシステムを作るという前提を仮定して，それに必要ないくつかのプログラムを，全体とのバランスを十分に考慮しつつ，考えていくことにしましょう。

ここでは，比較的簡単であり，利用度の高いという理由で，データ処理のシステムをとりあげることにしました。目標となる仮想システムは次のようなものにします。

1．データは10進数2桁以内の正の整数である。
2．データは，メモリにBCDコードで連続して入っているものとする。
3．データの合計は，9999以下であるものとし，それをBCDコードでメモリに納める。
4．データの平均は，小数点第3位を四捨五入し，第2位までをBCDコードで，メモリに納める。
5．データの順位づけを行なう。
6．データの分布状態を，5の幅で調べ21個のメモリにその度数を納める。

さて，目標が決まったところで，作業のおおまかな進め方を考えておきましょう。ここでは次のようなものにしてみました。

1．演算やカウントはすべて2進数で行なう。
2．1のために必要なBCD→2進変換，2進→BCD変換は，システムの最初と最後にまとめて行なう。
3．データやその合計が，目標で指定した範囲を越える場合の警告システムは含めない。
4．プログラムは次のブロックにわけて，まずは一つ一つのプログラムを制作する。

〔A〕 M(HL)に入っているデータをBCDコードで表わされているものとみなして，それを2進数に変換し，M(HL)に納めるプログラム。

〔B〕 M(HL)に入っている2進数をBCDコードに変換して，M(HL)に納めるプログラム。

〔C〕 二つのメモリM(DATA＋1)，M(DATA)にまたがって入っている2進数をBCDコードに変換して，M(DATA＋1)，M(DATA)に納めるプログラム。

〔D〕 連続してメモリに入っているデータ群の中で，個々のデータが何番目に大きいかを調べ，指定したメモリへその順位(ただし，2

進数）を納めるプログラム。

〔E〕 連続してメモリに入っている2進数のデータを5の幅で区切り，それぞれの範囲にいくつのデータがあるかを調べ，指定したメモリにその数(2進数)を納めるプログラム。

〔F〕 連続してメモリに2進数で入っているデータの和と2ワードにまたがる2進数で求め，指定したメモリへ納め，また平均を小数第2位(第3位を四捨五入)まで求め，整数部，小数部を分けて，別々のメモリに納めるプログラム。

以上のことをよく頭の中に入れて，上で述べた〔A〕～〔F〕までのプログラムを作っていくことにしましょう。

〔A〕 BCDコード→2進数変換プログラム

メモリに入っているデータをBCDコードと見なして，それを2進数に変換するプログラムです。メモリの番地は，HLレジスタペアで指定することにしました。連続したデータに対して，この処理を行なうときに効率よくできるからです。

ここで，BCDコードで表わされたデータを2進数に変換するアルゴリズムについて考えてみましょう。

たとえば10進数の99という数がBCDコードでメモリに入っていたとすると，そのメモリの上位4ビットが9を，下位4ビットが同じく9を表わしていることになります。したがって，メモリの各ビットは　10011001　となっているはずです。

これを10進数に直せば99に相当する2進数に変換するわけですから，各ビットを次のようにしなければなりません。

01100011

どうすればよいかといいますと，メモリに入っているBCDコードのデータの上位4ビットの1001をとりだし，これに10進数の10に相当する　1010　を掛けて，その結果と下位4ビットの　1001　を加えればよいのです。

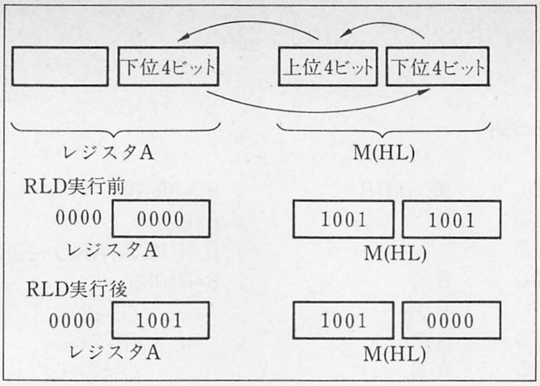
では，どうやって，BCDコードで表わされているデータの上位4ビットと下位4ビットを分けたらよいのでしょうか。

理論的には，8ビットのデータを4ビットずつに分けたければ，それを2進数10000で割って，商と余りにわければよいのですが，割り算の命令のないマイクロコンピュータでは，それは簡単な方法とはいえません。

シフト命令や，ローテイト命令を使って，4ビットずつに分ける方法もありますが，Z－80独特の命令RLD命令を使えばことは非常に簡単になります。

RLD(ROTATE, DIGIT　LEFT)は，レジスタAの下位4ビットと，M(HL)の上位4ビットとM(HL)の下位4ビットを**第14－1図**のように回転させる命令です。したがって，レジスタAをゼロにしておいてから，この命令を実行すれば，4ビットずつ取り出すことができるわけです。

次に，取り出したM(HL)の上位4ビットに2進数　1010　を掛けなければなりませんが，Z80に掛け算の命令はありません。とり出した4ビットの数を10回加えてもよいのですが，もう少しよ



《第14－1図》 RLD命令の復習

い方法があります。

数学でいう分配法則を使うのです。つまり，

〔上位4ビット〕×1010

＝ 〔上位4ビット〕×(1000＋10)

＝ 〔上位4ビット〕×(100＋1)×10

＝ {〔上位4ビット〕×10} ×(100＋1)

＝ {〔上位4ビット〕×10} ×100

＋ {〔上位4ビット〕×10}

ところで，10を掛けるということは2倍すること，100を掛けるということは4倍することに他なりません。したがって簡単に実行できるわけです。

たとえば，レジスタAに1010を掛ける場合なら，

① Aを2倍する（A←A＋A）

② このAをBにも入れておく

③ Aを4倍する（A←A＋Aを2回行なう）

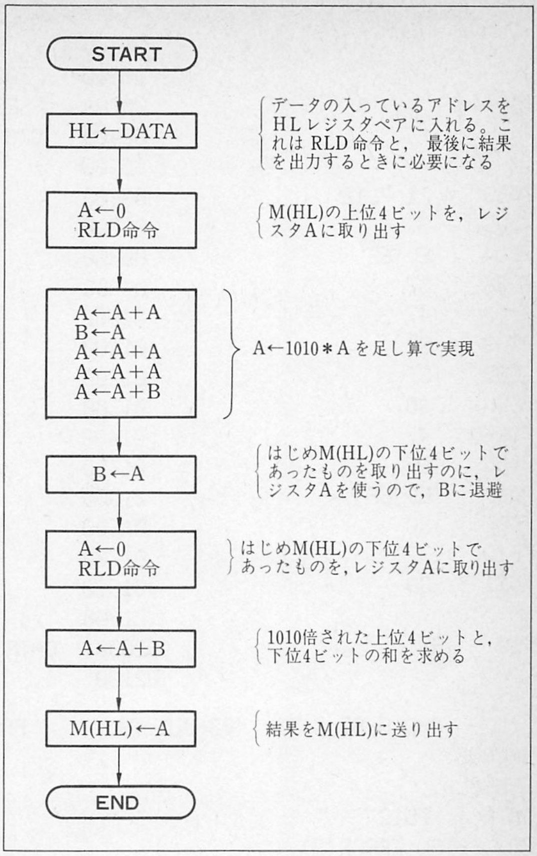
④ このAにBの内容を加わえるとすればＯＫです。

以上のことを利用して，メモリに入っているＢＣＤコードのデータを2進数に変換するプログラムのフローチャートを書いてみましょう。

第14－2図のようになりました。データの入っているメモリのアドレスをDATAとしました。ＲLD命令を実行する前に，レジスタAをゼロにしておくことは忘れがちです。注意しましょう。後は特に問題は無いでしょう。

このフローチャートをもとにアセンブラプログラミングをすると**第14－3図**のようになります。マイクロソフト社のMACRO－80を使ってアセンブルしたものですが，ラインナンバ 600のところのアドレスと機械結の順序には注意しなければいけません。

7D00　21　7D12



《第14－2図》M(HL) 内のデータをBCDコードとみなして，それを2進数に変換するプログラムのフローチャート

とありますが，一つ一つメモリに入れるときには，

M（7D00）＝21

M（7D01）＝12

M（7D02）＝7D

としなければなりません。実際には，MACRO－80を含むソフトウェア・パッケージを用いる場合には，このリストを見ながら，機械語を入力することはありませんから，アドレスなどを表わす2ワードにまたがる数値は，アドレス・シンボルとの対応を見やすくする方に重点が置かれ，このようになっているものと思われます。その点を頭に入れて注意をしておいてください。

また，ここで作られたプログラムを後で実際に動かしてみるときに便利なように，プログラムの最後はHALTではなく，RETにしてあります。

データは，とりあえずBCDコードで99としてあります。

第14－4図は，**第14－3図**から得られた機械語

```
     MACRO-80 3.31   03-JUL-79        PAGE    1
                                00100   ,       TRANSFORM  BCD INTO BINARY
                                00200   .Z80
 0000'                          00300   ASEG
                                00400   ORG     7D00H
                                00500   ;
 7D00     21 7D12               00600           LD      HL,DATA     ; HL=DATA
 7D03     97                    00700           SUB     A           ; A=0
 7D04     ED 6F                 00800           RLD                 ; A.M(HL)-M(HL)*10H
 7D06     87                    00900           ADD     A,A         , B=A+0AH
 7D07     47                    01000           LD      B,A         ;       (B=A)
 7D08     87                    01100           ADD     A,A         ;       (A=2*A)
 7D09     87                    01200           ADD     A,A         ;       (A=2*A)
 7D0A     80                    01300           ADD     A,B         ;       (A=A+B)
 7D0B     47                    01400           LD      B,A         ;       (B-A)
 7D0C     97                    01450           SUB     A           ; A=0
 7D0D     ED 6F                 01500           RLD                 ; A.M(HL)=M(HL)*10H
 7D0F     80                    01600           ADD     A,B         ; A-A+B
 7D10     77                    01700           LD      (HL),A      ; M(HL)=A
 7D11     C9                    01800           RET                 ; RETURN
                                01900   ,
7D12     99                     02000   DATA:   DB      99H         , M(DATA)=99H
                                02100           END

       MACRO-80 3.31   03-JUL-79        PAGE    S
 MACROS.
 SYMBOLS.
 DATA    7D12
 NO  FATAL ERROR(S)
```

《第14－3図》 第14－2図のフローチャートをもとにアセンブラプログラムしたもの

を，実際に７Ｄ00番地以降のメモリに一つずつ入れ，一つのサブルーチンとして動かしてみたときに，メモリがどのように変化したかを示すものです。NO.19のデータ，つまりＭ(７Ｄ12)の99が，実行後，63になっていることがわかります。これは，10011001　が10進数に直せば99になる２乗数　01100011　に変換されたことを物語っています。

第14－５図はDATAとして，10進数の15をとりあげて，今作ったプログラムがうまく働くかどうか調べてみたものです。NO.19の16進数を15にして，Ｍ(７Ｄ12)を　00010101　にしてから実行です。

実行後，Ｍ(７Ｄ12)が０Ｆつまり　00001111　になっていることがわかります。

以上で，プログラム〔Ａ〕は完成しました。

```
** BEFORE RUN
---------------------------------------
NO 1      7D00      21
NO 2      7D01      12
NO 3      7D02      7D
NO 4      7D03      97
NO 5      7D04      ED
NO 6      7D05      6F
NO 7      7D06      87
NO 8      7D07      47
NO 9      7D08      87
NO 10     7D09      87
NO 11     7D0A      80
NO 12     7D0B      47
NO 13     7D0C      97
NO 14     7D0D      ED
NO 15     7D0E      6F
NO 16     7D0F      80
NO 17     7D10      77
NO 18     7D11      C9
NO 19     7D12      99
---------------------------------------
** AFFTER RUN
---------------------------------------
NO 1      7D00      21
NO 2      7D01      12
NO 3      7D02      7D
NO 4      7D03      97
NO 5      7D04      ED
NO 6      7D05      6F
NO 7      7D06      87
NO 8      7D07      47
NO 9      7D08      87
NO 10     7D09      87
NO 11     7D0A      80
NO 12     7D0B      47
NO 13     7D0C      97
NO 14     7D0D      ED
NO 15     7D0E      6F
NO 16     7D0F      80
NO 17     7D10      77
NO 18     7D11      C9
NO 19     7D12      63
---------------------------------------
```

《第14－4図》 第14－3図の機械語を実際にメモリに入れ実行前後の変化をみたもの

```
** BEFORE RUN
---------------------------------------
NO 1      7D00      21
NO 2      7D01      12
NO 3      7D02      7D
NO 4      7D03      97
NO 5      7D04      ED
NO 6      7D05      6F
NO 7      7D06      87
NO 8      7D07      47
NO 9      7D08      87
NO 10     7D09      87
NO 11     7D0A      80
NO 12     7D0B      47
NO 13     7D0C      97
NO 14     7D0D      ED
NO 15     7D0E      6F
NO 16     7D0F      80
NO 17     7D10      77
NO 18     7D11      C9
NO 19     7D12      15
---------------------------------------
** AFFTER RUN
---------------------------------------
NO 1      7D00      21
NO 2      7D01      12
NO 3      7D02      7D
NO 4      7D03      97
NO 5      7D04      ED
NO 6      7D05      6F
NO 7      7D06      87
NO 8      7D07      47
NO 9      7D08      87
NO 10     7D09      87
NO 11     7D0A      80
NO 12     7D0B      47
NO 13     7D0C      97
NO 14     7D0D      ED
NO 15     7D0E      6F
NO 16     7D0F      80
NO 17     7D10      77
NO 18     7D11      C9
NO 19     7D12      0F
---------------------------------------
```

《第14－5図》 BCDコード00010101を２進数にに変換する実験

〔B〕 2進数をBCDコード交換するプログラム

前頁のプログラムAの逆を行なうプログラムを作るわけです。つまり，M(HL)に入っているデータを2進数であると見なして，それをBCDコードに変換して，M(HL)に納めるプログラムを作るわけです。

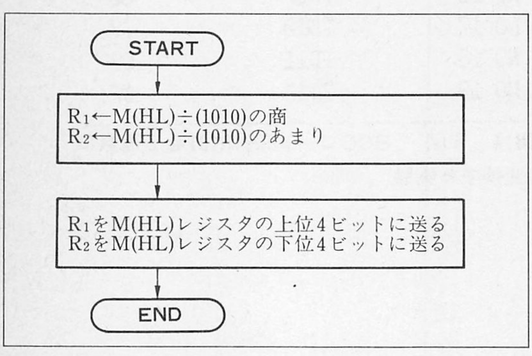
さて，このアルゴリズムは，どのようなものでしょうか。

この場合，10進数に直せば99以下のデータしか扱かいませんから，**第14－6図**のようなアルゴリズムで大丈夫です。

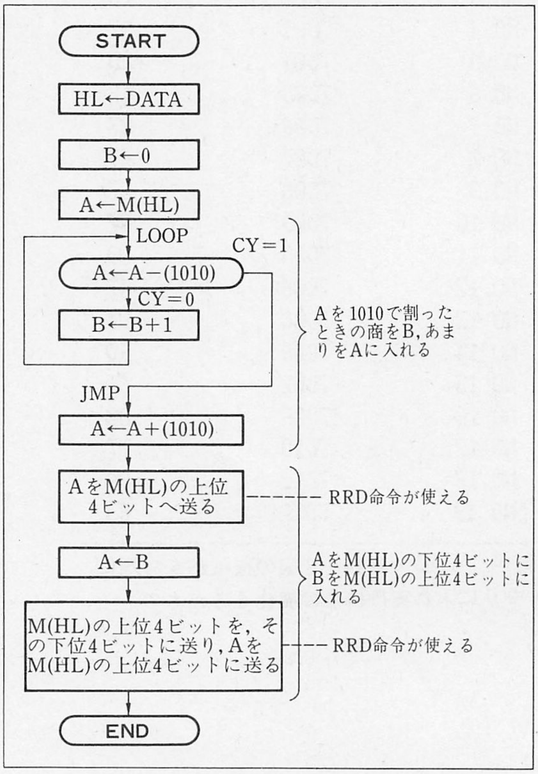
このアルゴリズムに従って行うものとするとZ80には，割り算の命令がありませんからまずはそこから考えなければなりません。

割り算は引き算の繰り返しで実現することにしたとしても，最大9回の繰り返しですみますから，実用的でないとはいえないでしょう。

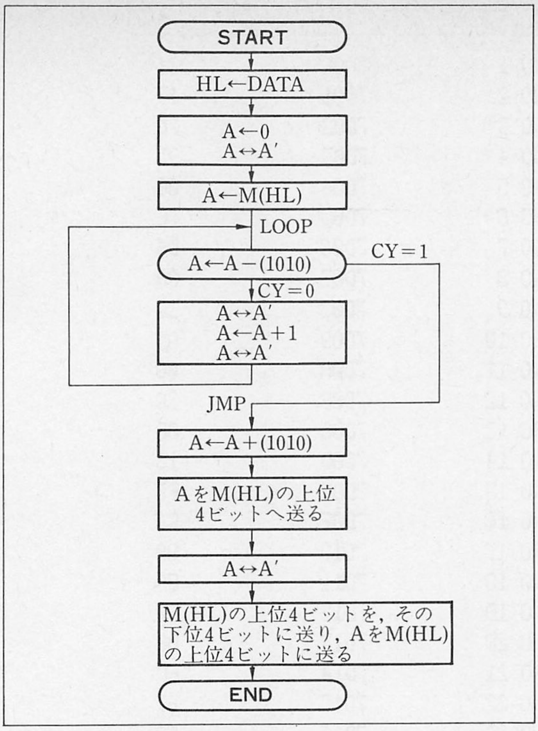
そこで，**第14－6図**のアルゴリズムを　Z80の命令を頭に入れて，書きかえてみることにしましょう。**第14－7図**のようになりました。これでよいのですが，ちょっと色気を出して，レジスタBを使わないでやる方法を工夫してみましょう。2ワードほどプログラムが長くなってしまいましたが，レジスタBを退避させなければいけない場合には，2ワード以上が必要になりますから，がまんできる範囲です。**第14－8図**が，そのフローチ



《第14－6図》　M(HL)内の2進数をBCDコードに変換するアルゴリズム



《第14－7図》　第14－6図のアルゴリズムをZ－80用に書きかえたもの



《第14－8図》 第14－7図のレジスタBをAの補助レジスタA′に代えたもの

ャートです。レジスタAの補助レジスタA′をBの代りに使ったわけです。

第14－8図のフローチャートをもとに，アセンブラプログラミングしたものが，**第14－9図**です。おもな注意点はプログラム〔A〕の場合と同じです。

ラインナンバ　1100，1600，2200　のところに，数字＋Dというのがありますが，これは，その数字が10進数であることを示すものです。もしHであれば，それは16進数，Bであれば2進数であることになります。

また，データは，プログラム〔A〕の場合のちょうど逆になるように，01100011　としました。

第14－10図は，**第14－9図**から得られた機械語を一つ一つメモリに入れ，実際に実行してみたときに，メモリの内容が目標どおり変化しているかどうか調べたものです。ここでNO.24のデータが63から99に変っているのが確認できます。

第14－11図は，データをゼロにした場合の確認を行なおうとしたものです。ゼロはゼロになっています。

以上でプログラム〔B〕は完成です。

```
     MACRO-80 3.31   03-JUL-79        PAGE    1
                            00100   ;          TRANSFORM BINARY INTO BCD
                            00200   .Z80
0000'                       00300   ASEG
                            00400   ORG        7D00H
                            00500   ;
7D00    21 7D17             00600              LD      HL,DATA        , HL=DATA
7D03    97                  00800              SUB     A              , A=0
7D04    08                  00900              EX      AF,AF'         , A<-->A'
7D05    7E                  01000              LD      A,(HL)         , A=M(HL)
7D06    D6 0A               01100   LOOP.      SUB     10D            , A=A-10D
7D08    38 05               01150              JR      C,JMP          , IF CY=1 GOTO JMP
7D0A    08                  01200              EX      AF,AF'         , A<-->A'
7D0B    3C                  01300              INC     A              , A=A+1
7D0C    08                  01400              EX      AF,AF'         , A<-->A'
7D0D    18 F7               01500              JR      LOOP           , GOTO LOOP
7D0F    C6 0A               01600   JMP.       ADD     A,10D          , A=A+10D
7D11    ED 67               01700              RRD                    ; ROTATE DIGIT RIGHT
7D13    08                  01800              EX      AF,AF'         , A<-->A'
7D14    ED 67               01900              RRD                    , ROTATE DIGIT RIGHT
7D16    C9                  02000              RET                    , RETURN
                            02100
7D17    63                  02200   DATA.      DB      99D            , M(DATA)=99D
                            02300              END

      MACRO-80 3.31   03-JUL-79        PAGE    S
MACROS.
SYMBOLS.
DATA    7D17    JMP     7D0F    LOOP    7D06
NO  FATAL ERROR(S)
```

《第14－9図》 第14－8図のフローチャートをもとにアセンブラプログラミングしたもの

```
** BEFORE RUN
--------------------------------------
NO 1        7D00        21
NO 2        7D01        17
NO 3        7D02        7D
NO 4        7D03        97
NO 5        7D04        08
NO 6        7D05        7E
NO 7        7D06        D6
NO 8        7D07        0A
NO 9        7D08        38
NO 10       7D09        05
NO 11       7D0A        08
NO 12       7D0B        3C
NO 13       7D0C        08
NO 14       7D0D        18
NO 15       7D0E        F7
NO 16       7D0F        C6
NO 17       7D10        0A
NO 18       7D11        ED
NO 19       7D12        67
NO 20       7D13        08
NO 21       7D14        ED
NO 22       7D15        67
NO 23       7D16        C9
NO 24       7D17        63
--------------------------------------
** AFFTER RUN
--------------------------------------
NO 1        7D00        21
NO 2        7D01        17
NO 3        7D02        7D
NO 4        7D03        97
NO 5        7D04        08
NO 6        7D05        7E
NO 7        7D06        D6
NO 8        7D07        0A
NO 9        7D08        38
NO 10       7D09        05
NO 11       7D0A        08
NO 12       7D0B        3C
NO 13       7D0C        08
NO 14       7D0D        18
NO 15       7D0E        F7
NO 16       7D0F        C6
NO 17       7D10        0A
NO 18       7D11        ED
NO 19       7D12        67
NO 20       7D13        08
NO 21       7D14        ED
NO 22       7D15        67
NO 23       7D16        C9
NO 24       7D17        99
--------------------------------------
```

《第14−10図》　第14−9図の機械語をメモリに入れ，実行前後のメモリの変化をみたもの

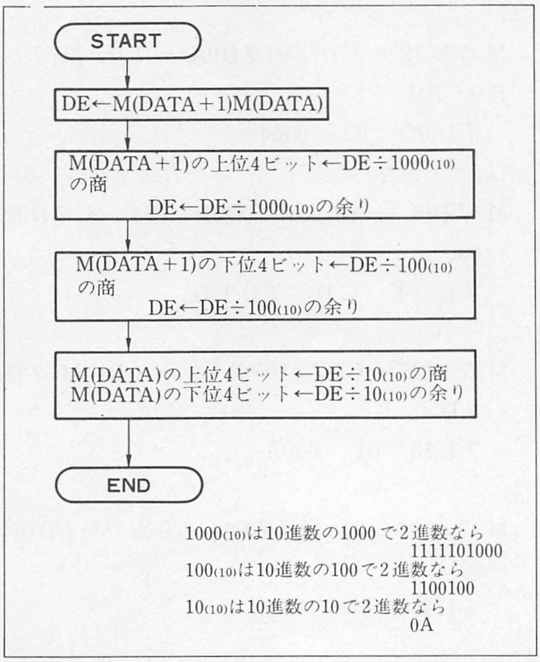
```
** BEFORE RUN
--------------------------------------
NO 1        7D00        21
NO 2        7D01        17
NO 3        7D02        7D
NO 4        7D03        97
NO 5        7D04        08
NO 6        7D05        7E
NO 7        7D06        D6
NO 8        7D07        0A
NO 9        7D08        38
NO 10       7D09        05
NO 11       7D0A        08
NO 12       7D0B        3C
NO 13       7D0C        08
NO 14       7D0D        18
NO 15       7D0E        F7
NO 16       7D0F        C6
NO 17       7D10        0A
NO 18       7D11        ED
NO 19       7D12        67
NO 20       7D13        08
NO 21       7D14        ED
NO 22       7D15        67
NO 23       7D16        C9
NO 24       7D17        00
--------------------------------------
** AFFTER RUN
--------------------------------------
NO 1        7D00        21
NO 2        7D01        17
NO 3        7D02        7D
NO 4        7D03        97
NO 5        7D04        08
NO 6        7D05        7E
NO 7        7D06        D6
NO 8        7D07        0A
NO 9        7D08        38
NO 10       7D09        05
NO 11       7D0A        08
NO 12       7D0B        3C
NO 13       7D0C        08
NO 14       7D0D        18
NO 15       7D0E        F7
NO 16       7D0F        C6
NO 17       7D10        0A
NO 18       7D11        ED
NO 19       7D12        67
NO 20       7D13        08
NO 21       7D14        ED
NO 22       7D15        67
NO 23       7D16        C9
NO 24       7D17        00
--------------------------------------
```

《第14−11図》　データを0にして実行してみたもの

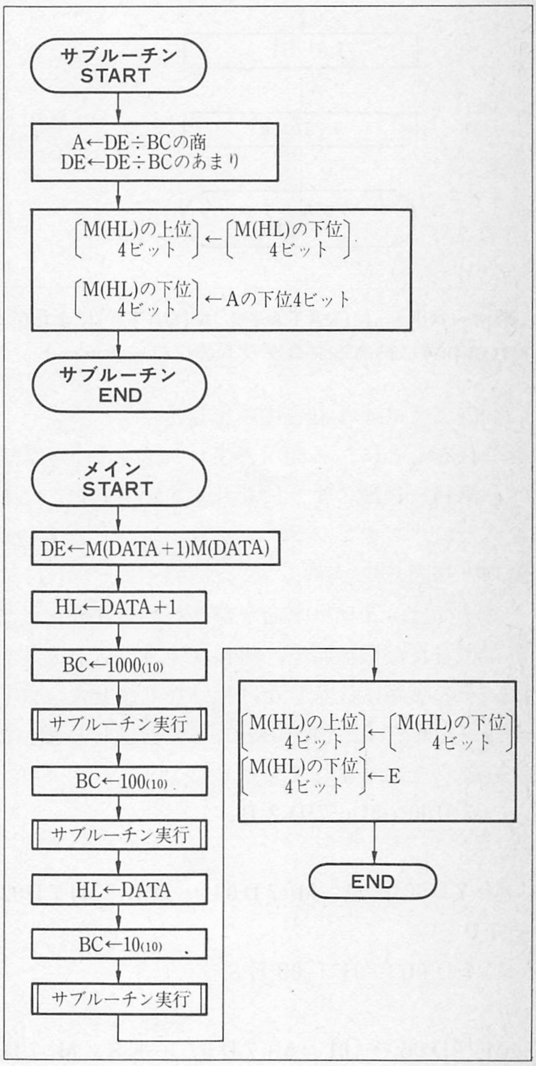
〔C〕 2ワードにまたがって入っている2進法をBCDコードに変換するプログラム

M(DATA＋1)M(DATA)にまたがって入っている2進数をBCDコードに変換して，M(DATA＋1)M(DATA)に納めるプログラムですが，変換のアルゴリズムの基本は，プログラム〔B〕と同じです。それを**第14－12図**にまとめてみました。

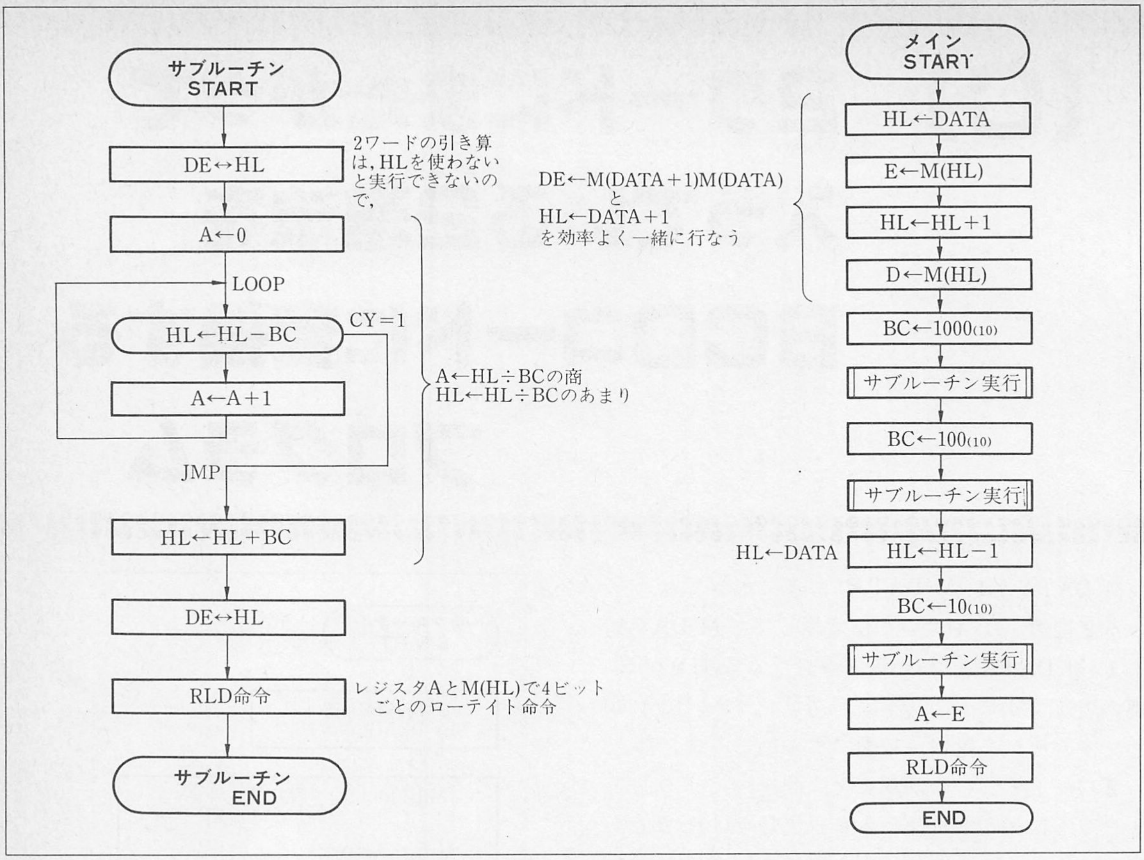
2ワードどうしの割り算と，商をメモリの上位4ビットに送るプログラムを工夫しなければなりませんが，これらがほぼ同じ形で3回も顔を出すことに注目して，サブルーチンの形にすることにしましょう。そのために，**第14－12図**を，**第14－**



《第14－12図》　2ワードにまたがって入っている2進数をBCDコードに変換するアルゴリズム



《第14－13図》　第14－12図のフローチャートをいくぶん具体化させたもの



《第14－14図》 M(DATA＋1)M(DATA)にまたがって入っている2進数をBCDコードに変換してM(DATA＋1)M(DATA)に納めるプログラムのフローチャート

13図のように書き直してみました。

これをもとに，Z80用のフローチャートを書くと，**第14－14図**のようになります。さらに，これをもとに，アセンブラプログラミングしたものが**第14－15図**です。

ここでは，3ワード命令が多いのですが，ここから得られた機械語を，直接メモリに入れる場合は，その順序に注意しなければなりません。詳しくは，プログラム〔A〕のところでお話ししましたが，要点を繰り返しておきましょう。

7D00　21　7D2B

は，

M(7D00)←21，M(7D01)←2B，M(7D02)←7D

7D06　01　03E8

は，

M(7D06)←01，M(7D07)←E8，M(7D08)←03

7D09　CD　7D1D

は，

M(7D09)←CD，M(7D0A)←1D，M(7D0B)←7D

7D0C　01　0064

は，

M(7D0C)←01，M(7D0D)←64，M(7D0E)←00

7D0F　CD　7D1D

は，

M(7D0F)←CD，M(7D10)←1D，M(7D11)←7D

7D13　01　000A

は，

M(7D13)←01，M(7D14)←0A，M(7D15)←00

7D16　CD　7D1D

は，

M(7D16)←CD，M(7D17)←1D，M(7D18)←7D

```
  MACRO-80 3.31   03-JUL-79       PAGE    1
                  ;       TRANSFORM BINARY INTO BCD(16-BIT)
                  .Z80
0000'             ASEG
                  ORG     7D00H
                  ;
7D00    21 7D2B           LD      HL,DATA         ; DE-M(DATA+1)M(DATA)
7D03    5E                LD      E,(HL)
7D04    23                INC     HL              ;       (HL-DATA+1)
7D05    56                LD      D,(HL)
                  ;
7D06    01 03E8           LD      BC,1000D        ; BC-1000D
7D09    CD 7D1D           CALL    SRT             ; GOSUB SRT
7D0C    01 0064           LD      BC,100D         ; BC-100D
7D0F    CD 7D1D           CALL    SRT             ; GOSUB SRT
                  ;
7D12    2B                DEC     HL              ; HL-DATA
7D13    01 000A           LD      BC,10D          ; BC-10D
7D16    CD 7D1D           CALL    SRT             ; GOSUB SRT
7D19    7B                LD      A,E             ; A-E
7D1A    ED 6F             RLD                     ; ROTET DIGIT LEFT
7D1C    C9                RET                     ; END
                  ;
                  ;--------- SUBROUTINE SRT --------------
7D1D    EB        SRT.    EX      DE,HL           ; DE<-->HL
7D1E    97                SUB     A               ; A-0 , CY-0
7D1F    ED 42     LOOP.   SBC     HL,BC           ; HL-HL-BC-CY
7D21    38 03             JR      C,JMP           ; IF CY-1 GOTO JMP
7D23    3C                INC     A               ; A-A+1
7D24    18 F9             JR      LOOP            ; GOTO LOOP
7D26    09        JMP.    ADD     HL,BC           ; HL-HL+BC
7D27    EB                EX      DE,HL           ; DE<-->HL
7D28    ED 6F             RLD                     ; ROTATE DIGIT LEFT
7D2A    C9                RET                     ; GOTO MAIN
                  ;---------------------------------------------
                  ;
7D2B    E7        DATA.   DB      0E7H            ; M(DATA+1)M(DATA)-03E7H
7D2C    03                DB      03H
                          END

 MACRO-80 3.31   03-JUL-79       PAGE    S
MACROS.
SYMBOLS.
DATA    7D2B    JMP     7D26    LOOP    7D1F    SRT     7D1D
NO  FATAL ERROR(S)
```

《第14−15図》 M(DATA＋１)M(DATA)にまたがって入っている２進数をＢＣＤコードに変換してM(DATA＋１)M(DATA)に納めるプログラム

となります。

ここでは，データを10進数に直せば 999になる２進数，1111100111 としておきました。したがって

M(DATA＋１)←３H
M(DATA)　←0E７H

を入れておきます。

プログラムが正常に働けば，二つのメモリの内

```
** BEFORE RUN
----------------------------------------
NO 1        7D00       21
NO 2        7D01       2B
NO 3        7D02       7D
NO 4        7D03       5E
NO 5        7D04       23
NO 6        7D05       56
NO 7        7D06       01
NO 8        7D07       E8
NO 9        7D08       03
NO 10       7D09       CD
NO 11       7D0A       1D
NO 12       7D0B       7D
NO 13       7D0C       01
NO 14       7D0D       64
NO 15       7D0E       00
NO 16       7D0F       CD
NO 17       7D10       1D
NO 18       7D11       7D
NO 19       7D12       2B
NO 20       7D13       01
NO 21       7D14       0A
NO 22       7D15       00
NO 23       7D16       CD
NO 24       7D17       1D
NO 25       7D18       7D
NO 26       7D19       7B
NO 27       7D1A       ED
NO 28       7D1B       6F
NO 29       7D1C       C9
NO 30       7D1D       EB
NO 31       7D1E       97
NO 32       7D1F       ED
NO 33       7D20       42
NO 34       7D21       38
NO 35       7D22       03
NO 36       7D23       3C
NO 37       7D24       18
NO 38       7D25       F9
NO 39       7D26       09
NO 40       7D27       EB
NO 41       7D28       ED
NO 42       7D29       6F
NO 43       7D2A       C9
NO 44       7D2B       E7
NO 45       7D2C       03
----------------------------------------
```

```
** AFFTER RUN
----------------------------------------
NO 1        7D00       21
NO 2        7D01       2B
NO 3        7D02       7D
NO 4        7D03       5E
NO 5        7D04       23
NO 6        7D05       56
NO 7        7D06       01
NO 8        7D07       E8
NO 9        7D08       03
NO 10       7D09       CD
NO 11       7D0A       1D
NO 12       7D0B       7D
NO 13       7D0C       01
NO 14       7D0D       64
NO 15       7D0E       00
NO 16       7D0F       CD
NO 17       7D10       1D
NO 18       7D11       7D
NO 19       7D12       2B
NO 20       7D13       01
NO 21       7D14       0A
NO 22       7D15       00
NO 23       7D16       CD
NO 24       7D17       1D
NO 25       7D18       7D
NO 26       7D19       7B
NO 27       7D1A       ED
NO 28       7D1B       6F
NO 29       7D1C       C9
NO 30       7D1D       EB
NO 31       7D1E       97
NO 32       7D1F       ED
NO 33       7D20       42
NO 34       7D21       38
NO 35       7D22       03
NO 36       7D23       3C
NO 37       7D24       18
NO 38       7D25       F9
NO 39       7D26       09
NO 40       7D27       EB
NO 41       7D28       ED
NO 42       7D29       6F
NO 43       7D2A       C9
NO 44       7D2B       99
NO 45       7D2C       09
----------------------------------------
```

《第14－16図》 第14－15図から得られた機械語を直接メモリに入れ実行前後の内容の変化をみたもの

容は，

M(DATA＋1)＝09H

M(DATA)　＝99H

となるはずです。

第14－16図は，**第14－15図**から得られた機械語を，直接，7D00番地以降のメモリに入れ，実際に動かしてみたときに，メモリの内容がどう変るかを見たものです。

実行前に，E7，03であったNO.44とNO.45の内容は，実行後，99，09になっていることが確認できます。

これで，プログラム〔C〕は完成です。

〔D〕 順位づけのプログラム

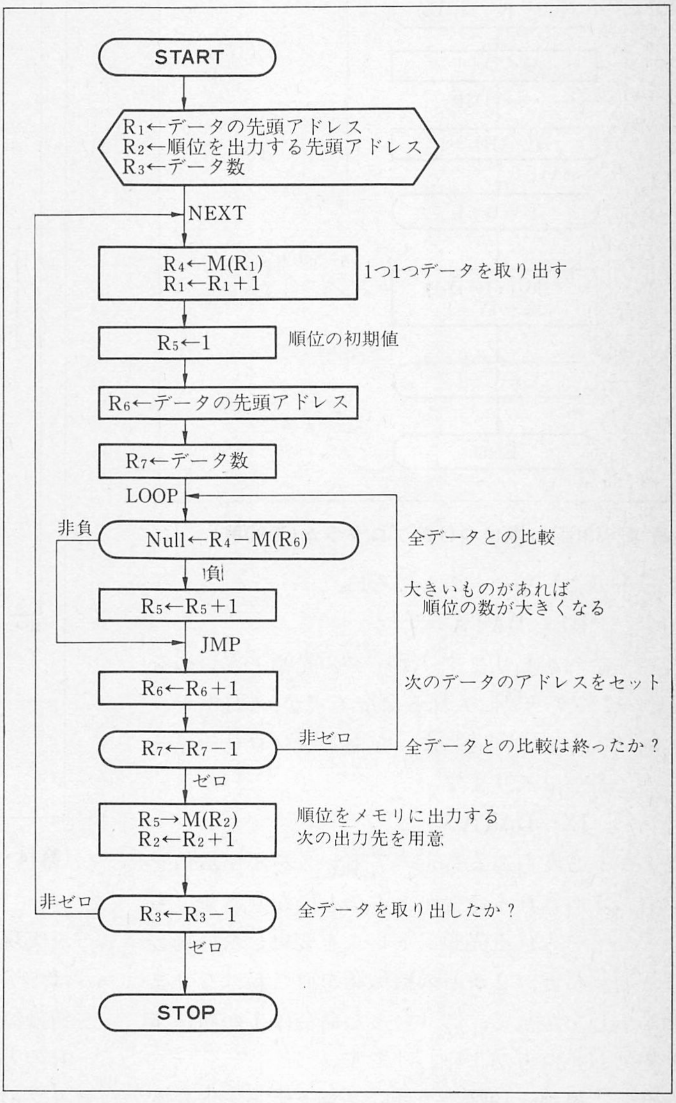
連続してメモリに入っているデータ群の中で，個々のデータが何番目に大きいか，を調べ，その順位を指定したメモリへ納めるプログラムです。

プログラム実行前に，データ群の先頭アドレスと，データ数と，順位を納めるメモリの先頭アドレスが与えられるものとして，アルゴリズムを考えてみましょう。

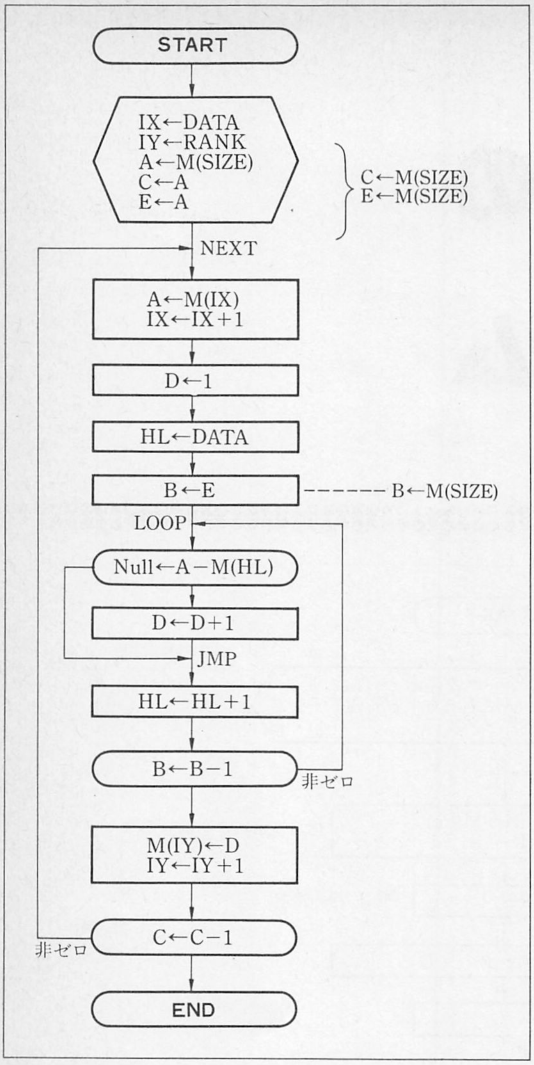
おおまかには，一つ一つのデータを取り出し，それを全部のデータを比較して，大きなものがいくつあったかを調べ，それに1を加わえたものが，そのデータの全体における順位である。と考えればよいのです。これをフローチャートに表わすと，**第14－17図**のようになります。

さて，**第14－17図**でＲ１からＲ７で表わされているレジスタを，Ｚ80用には，どのレジスタにわりつけたらよいでしょうか。それの良し悪しで，プログラムの良し悪しが決ってしまいますここでは，**第14－18図**のようにしてみました。ただし，そこでは，データの先頭アドレスをDATA，順位出力先の先頭アドレスをRANK，データ数の入っているメモリのアドレスをSIZEとしてあります。

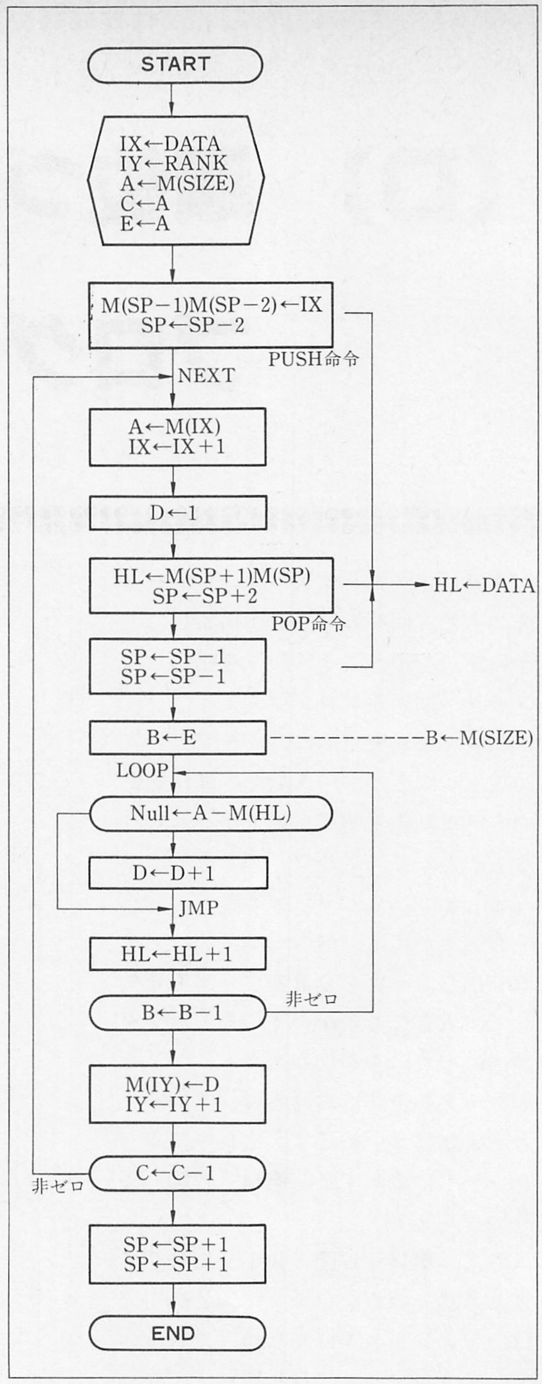
これでよいのですが，フローチャー



《**第14－17図**》 **順位づけのアルゴリズム**



《第14−18図》 順位づけのプログラム(Z80用)



《第14−19図》 第14−18図を改造したもの

トの中央よりやや上のところに，

HL←DATA

という命令があります。データの先頭アドレスをHLレジスタペアに入れる命令ですが，これがどうも気になります。同じような命令がフローチャートの一番上にあります。

IX←DATA

ですが，2度もある特定のアドレスを示す数値をレジスタに入れる命令があることになります。もしデータを入れる先頭アドレスを変更したい場合に，これだと，2か所の機械語を直さねばなりません。したがって，こういった命令は1か所にまとめておいた方がよいわけです。

そこで**第14−19図**のようにフローチャートを変えてみました。PUSH命令，POP命令を使って，

HL←DATA

を実現しようとしたものです。PUSH, POP命令はペアで用いればSPつまり，スタックポインタの値は気にしなくてもよいのですが，ここでは，POPの方がループの内側にあるため，スタックポインタのバランスがくずれてしまいます。そこで，2か所でその値を調節することにしてあります。

```
     MACRO-80 3.31   03-JUL-79        PAGE    1
                                 00100  ;       RANKING
                                 00200  .Z80
0000'                            00300  ASEG
                                 00400  ORG     7D00H
                                 00500  ;
7D00     DD 21 7D2D              00600          LD      IX,DATA         ; IX-DATA
7D04     FD 21 7D32              00700          LD      IY,RANK         ; IY-RANK
7D08     3A 7D2C                 00800          LD      A,(SIZE)        ; C-M(SIZE), E-M(SIZE)
7D0B     4F                      00900          LD      C,A
7D0C     5F                      01000          LD      E,A
7D0D     DD E5                   01100          PUSH    IX              ; M(SP-1)M(SP-2)-IX,SP-SP-2
                                 01200  ;
7D0F     DD 7E 00                01300  NEXT:   LD      A,(IX+0)        ; A-M(IX)
7D12     DD 23                   01400          INC     IX              ; IX-IX+1
7D14     16 01                   01500          LD      D,1             ; D-1
7D16     E1                      01600          POP     HL              ; HL-M(SP+1)M(SP),SP-SP+2
7D17     3B                      01700          DEC     SP              ; SP-SP-2
7D18     3B                      01800          DEC     SP
7D19     43                      01900          LD      B,E             ; B-M(SIZE)
7D1A     BE                      02000  LOOP:   CP      (HL)            ; NULL-A-M(HL)
7D1B     30 01                   02100          JR      NC,JMP          ; IF CY-0 GOTO JMP
7D1D     14                      02200          INC     D               ; D-D+1
7D1E     23                      02300  JMP:    INC     HL              ; HL-HL+1
7D1F     10 F9                   02400          DJNZ    LOOP            ; B-B-1,IF B<>0 GOTO LOOP
                                 02500  ;
7D21     FD 72 00                02600          LD      (IY),D          ; M(IY)-D
7D24     FD 23                   02700          INC     IY              ; IY-IY+1
7D26     0D                      02800          DEC     C               ; C-C-1
7D27     20 E6                   02900          JR      NZ,NEXT         ; IF C<>0 GOTO NEXT
                                 03000  ;
7D29     33                      03100          INC     SP              ; SP-SP+2
7D2A     33                      03200          INC     SP
7D2B     C9                      03300          RET                     ; END
                                 03400  ;
7D2C     05                      03500  SIZE:   DB      5D              ; M(SIZE)-5
7D2D     64                      03600  DATA:   DB      100D
7D2E     1E                      03700          DB      30D
7D2F     4B                      03800          DB      75D
7D30     00                      03900          DB      0
7D31     2D                      04000          DB      45
7D32     00                      04100  RANK:   DB      0
7D33     00                      04200          DB      0
7D34     00                      04300          DB      0
7D35     00                      04400          DB      0
7D36     00                      04500          DB      0
                                 04600          END
```

《第14−20図》 順位づけのプログラム

```
      MACRO-80 3.31   03-JUL-79       PAGE    S
MACROS:
SYMBOLS:
DATA    7D2D    JMP     7D1E    LOOP    7D1A    NEXT    7D0F
RANK    7D32    SIZE    7D2C
NO  FATAL ERROR(S)
```

** BEFORE RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	2D
NO 4	7D03	7D
NO 5	7D04	FD
NO 6	7D05	21
NO 7	7D06	32
NO 8	7D07	7D
NO 9	7D08	3A
NO 10	7D09	2C
NO 11	7D0A	7D
NO 12	7D0B	4F
NO 13	7D0C	5F
NO 14	7D0D	DD
NO 15	7D0E	E5
NO 16	7D0F	DD
NO 17	7D10	7E
NO 18	7D11	00
NO 19	7D12	DD
NO 20	7D13	23
NO 21	7D14	16
NO 22	7D15	01
NO 23	7D16	E1
NO 24	7D17	3B
NO 25	7D18	3B
NO 26	7D19	43
NO 27	7D1A	BE
NO 28	7D1B	30
NO 29	7D1C	01
NO 30	7D1D	14
NO 31	7D1E	23
NO 32	7D1F	10
NO 33	7D20	F9
NO 34	7D21	FD
NO 35	7D22	72
NO 36	7D23	00
NO 37	7D24	FD
NO 38	7D25	23
NO 39	7D26	0D
NO 40	7D27	20
NO 41	7D28	E6
NO 42	7D29	33
NO 43	7D2A	33
NO 44	7D2B	C9
NO 45	7D2C	05
NO 46	7D2D	64
NO 47	7D2E	1E
NO 48	7D2F	4B
NO 49	7D30	00
NO 50	7D31	2D
NO 51	7D32	00
NO 52	7D33	00
NO 53	7D34	00
NO 54	7D35	00
NO 55	7D36	00

** AFFTER RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	2D
NO 4	7D03	7D
NO 5	7D04	FD
NO 6	7D05	21
NO 7	7D06	32
NO 8	7D07	7D
NO 9	7D08	3A
NO 10	7D09	2C
NO 11	7D0A	7D
NO 12	7D0B	4F
NO 13	7D0C	5F
NO 14	7D0D	DD
NO 15	7D0E	E5
NO 16	7D0F	DD
NO 17	7D10	7E
NO 18	7D11	00
NO 19	7D12	DD
NO 20	7D13	23
NO 21	7D14	16
NO 22	7D15	01
NO 23	7D16	E1
NO 24	7D17	3B
NO 25	7D18	3B
NO 26	7D19	43
NO 27	7D1A	BE
NO 28	7D1B	30
NO 29	7D1C	01
NO 30	7D1D	14
NO 31	7D1E	23
NO 32	7D1F	10
NO 33	7D20	F9
NO 34	7D21	FD
NO 35	7D22	72
NO 36	7D23	00
NO 37	7D24	FD
NO 38	7D25	23
NO 39	7D26	0D
NO 40	7D27	20
NO 41	7D28	E6
NO 42	7D29	33
NO 43	7D2A	33
NO 44	7D2B	C9
NO 45	7D2C	05
NO 46	7D2D	64
NO 47	7D2E	1E
NO 48	7D2F	4B
NO 49	7D30	00
NO 50	7D31	2D
NO 51	7D32	01
NO 52	7D33	04
NO 53	7D34	02
NO 54	7D35	05
NO 55	7D36	03

《第14－21図》 第14－20図から得られた機械語をメモリに入れ実行前後内容の変化をみたもの

```
** AFFTER RUN
-------------------------------------
NO 1        7D00      DD
NO 2        7D01      21
NO 3        7D02      2D
NO 4        7D03      7D
NO 5        7D04      FD
NO 6        7D05      21
NO 7        7D06      32
NO 8        7D07      7D
NO 9        7D08      3A
NO 10       7D09      2C
NO 11       7D0A      7D
NO 12       7D0B      4F
NO 13       7D0C      5F
NO 14       7D0D      DD
NO 15       7D0E      E5
NO 16       7D0F      DD
NO 17       7D10      7E
NO 18       7D11      00
NO 19       7D12      DD
NO 20       7D13      23
NO 21       7D14      16
NO 22       7D15      01
NO 23       7D16      E1
NO 24       7D17      3B
NO 25       7D18      3B
NO 26       7D19      43
NO 27       7D1A      BE
NO 28       7D1B      30
NO 29       7D1C      01
NO 30       7D1D      14
NO 31       7D1E      23
NO 32       7D1F      10
NO 33       7D20      F9
NO 34       7D21      FD
NO 35       7D22      72
NO 36       7D23      00
NO 37       7D24      FD
NO 38       7D25      23
NO 39       7D26      0D
NO 40       7D27      20
NO 41       7D28      E6
NO 42       7D29      33
NO 43       7D2A      33
NO 44       7D2B      C9
NO 45       7D2C      05
NO 46       7D2D      64
NO 47       7D2E      64
NO 48       7D2F      64
NO 49       7D30      64
NO 50       7D31      01
NO 51       7D32      01
NO 52       7D33      01
NO 53       7D34      01
NO 54       7D35      01
NO 55       7D36      05
-------------------------------------
```

《第14－22図》 同値の値のデータがある場合の順位づけの確認

この**第14－19図**をもとに，アセンブラプログラミングしたものが**第14—20図**です。機械語のリストの見方における注意点は，今までと同じです。以下の４ワード命令，３ワード命令には十分注意してください。

7 D00　DD　21　7 D 2 D

7 D04　FD　21　7 D32

7 D08　3 A　7 D 2 C

なお，

7 D 0 F　DD　7 E　00

7 D21　FD　72　00

は，表示の順と機械語の順は同じです。

ここでは，データ数は５，データは，10進数に直せば，100，30，75，0，45，になるような２進数にしてあります。プログラムが正常に働けば，RANKというシンボルのつけられた７D32以下のメモリに，100，30，75，0，45，の全体における順位が，データの順に対応して入るはずです。つまり，100の順位は１なので，M(７D32)に１が次の30というデータの順位は４なので，M(７D33)に４が，という具合にM(７D34)に２，M(7D35)に５，M(７D36)に３が入るはずです。

これを確かめるために，**第14－20図**から得られた機械語を一つ一つメモリに入れ，実行前と実行後のメモリの内容を見ようとしたものが**第14－21図**です。NO.51からNO.55のデータで，上で述べたことが確認できます。

参考のために，同値のデータがいくつもある場合の順位づけが正常に働くかどうかをチェックしたときの結果を**第14－22図**にあげておきます。四つのデータが同値で同順位，他の一つは順位が５になっていることが確認できます。

以上で，プログラム〔D〕は完成です。

〔E〕 度数分布表を作るプログラム

連続してメモリに入っているデータ群を，0～4，5～9，10～14，15～19……95～99,100以上の21の範囲に分類し，それぞれの範囲に納まるデータの数を別に用意した21のメモリに納めるプログラムを作るわけです。

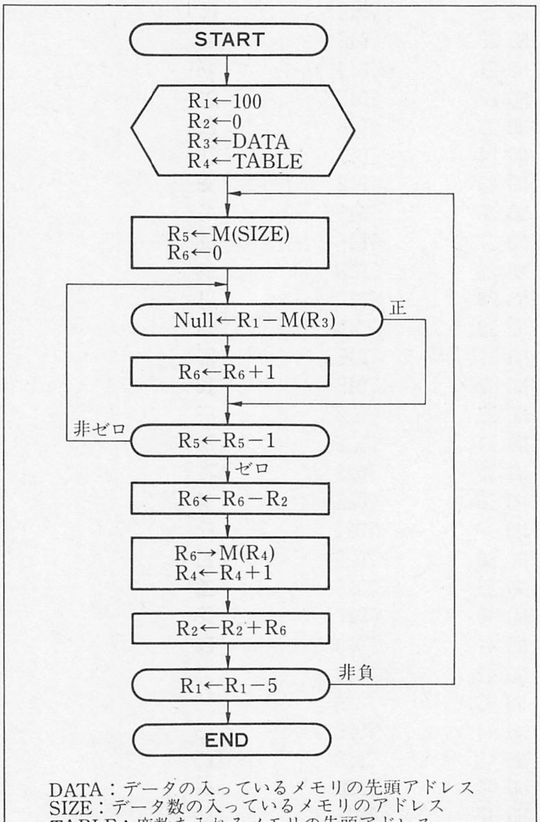
このアルゴリズムは二つのタイプがあります。一つは，21個のフルイを作り，全データをそれにかけ，どのフルイにかかったかで対応するメモリに1を加算していくタイプ。

もう一つは，一つ一つのデータから直接，演算などを通して，加算していくべきメモリのアドレスを求め，そのメモリに1を加わえていくタイプです。

前者のタイプをマイクロコンピュータ用のアルゴリズムで表わしたものが**第14－23図**です。100以上のデータの数，95以上のデータの数，90以上のデータの数を調べ，そこから5の幅の度数を求めているわけです。割り算の命令が不必要なことと，あらかじめ度数の入るメモリをクリアしておく必要のないことが特徴かもしれません。

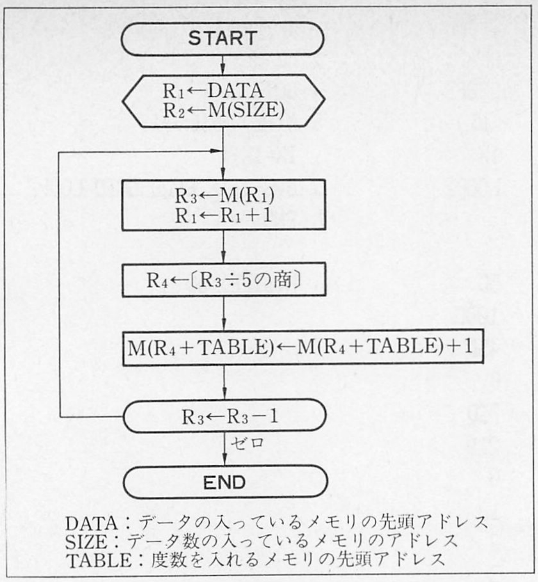
後者のタイプのアルゴリズムをマイクロコンピュータ用にしたものが**第14－24図**です。割り算の命令をもったマイクロコンピュータにはむいていますが，度数を入れるメモリをあらかじめクリアしておかなければなりません。

Z80用には，あえて後者のタイプのアルゴリズムを用いてみることにしました。**第14－24図**をZ80用に書きなおしたものが**第14－25図**です。後で独立したサブルーチンとしても使い易いように，

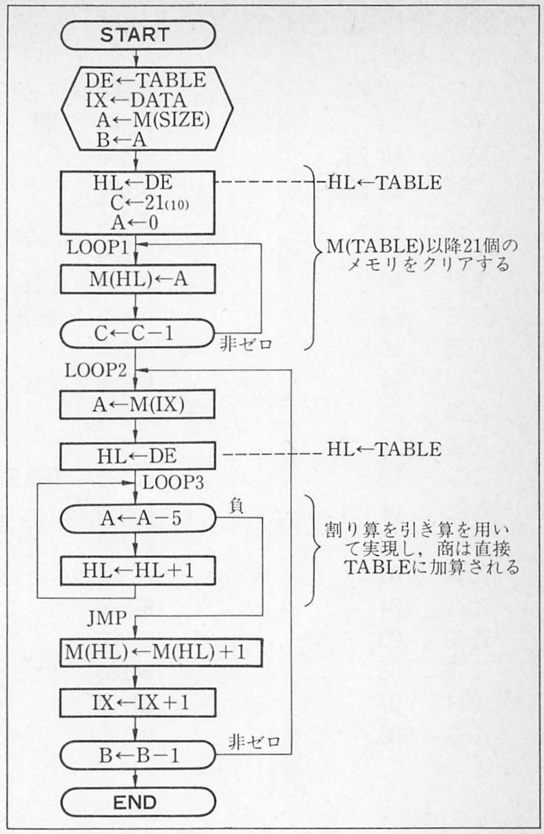


《第14－23図》度数分布表を作るアルゴリズム　タイプⅠ(100以上の度数，95～99の度数90～94の度数の順でM(TABLE)以降に入る)

入力出先のアドレス，TABLE，DATA，SIZE，は初めの部分だけにしか顔を出さないように配慮されています。そのためのDEレジスタペアの使い方に注意してください。



《第14−24図》度数分布表を作るアルゴリズム　タイプII（0～4の度数，5～92度数，10～14の度数の順でM(TABLE)以降のメモリに入る）



《第14−25図》　第14−24図をZ80用に書きなおしたもの

第14−25図のフローチャートをもとに，アセンブラプログラミングしたものが**第14−26図**です。リストを見る上での注意点は，前にお話しした通りです。16進数が切目なく4個つながって表示されている所は要注意です。

```
         MACRO-80 3.31   03-JUL-79        PAGE    1
                                  00100   ;       FREQUENCY TABLE
                                  00200   .Z80
  0000'                           00300   ASEG
                                  00400   ORG     7D00H
                                  00500   ;
  7D00     11 7D2D                00600           LD      DE,TABLE        ; DE=TABLE
  7D03     DD 21 7D28             00700           LD      IX,DATA         ; IX=DATA
  7D07     3A 7D27                00800           LD      A,(SIZE)        ; B=M(SIZE)
  7D0A     47                     00900           LD      B,A
                                  01000   ;
  7D0B     62                     01100           LD      H,D             ; HL=DE
  7D0C     6B                     01200           LD      L,E
  7D0D     97                     01300           SUB     A               ; A=0
  7D0E     0E 15                  01350           LD      C,21D           ; C=21D
  7D10     77                     01400   LOOP1.  LD      (HL),A          ; M(HL)=A
  7D11     23                     01450           INC     HL              ; HL=HL+1
  7D12     0D                     01500           DEC     C               ; C=C-1
  7D13     20 FB                  01600           JR      NZ,LOOP1        ; IF C<>0 GOTO LOOP1
                                  01700   ;
  7D15     DD 7E 00               01800   LOOP2.  LD      A,(IX)          ; A=M(IX)
  7D18     62                     01900           LD      H,D             ; HL=DE
  7D19     6B                     02000           LD      L,E
```

```
7D1A   D6 05       02100  LOOP3.  SUB    5D       , A=A-5
7D1C   38 03       02200          JR     C,JMP    , IF A<0 GOTO JMP
7D1E   23          02300          INC    HL       , HL=HL+1
7D1F   18 F9       02350          JR     LOOP3    , GOTO LOOP3
7D21   34          02400  JMP:    INC    (HL)     , M(HL)=M(HL)+1
7D22   DD 23       02500          INC    IX       , IX=IX+1
7D24   10 EF       02600          DJNZ   LOOP2    , B=B-1,IF B<>0 GOTO LOOP2
7D26   C9          02700          RET             , END
                   02800  ,
7D27   05          02900  SIZE:   DB     5D       , M(SIZE)=5D
7D28   64          03000  DATA..  DB     100D
7D29   2D          03100          DB     45D
7D2A   00          03200          DB     0
7D2B   49          03300          DB     73D
7D2C   15          03400          DB     21D
7D2D   00          03500  TABLE.  DB     0
7D2E   01          03600          DB     1
7D2F   02          03700          DB     2
7D30   03          03800          DB     3
7D31   04          03900          DB     4
7D32   05          04000          DB     5
7D33   06          04100          DB     6
7D34   07          04200          DB     7
7D35   08          04300          DB     8H
```

《第14－26図・その１》 度数分布表を作るプログラム

```
** BEFORE RUN
---------------------------------------
NO 1     7D00    11
NO 2     7D01    2D
NO 3     7D02    7D
NO 4     7D03    DD
NO 5     7D04    21
NO 6     7D05    28
NO 7     7D06    7D
NO 8     7D07    3A
NO 9     7D08    27
NO 10    7D09    7D
NO 11    7D0A    47
NO 12    7D0B    62
NO 13    7D0C    6B
NO 14    7D0D    97
NO 15    7D0E    0E
NO 16    7D0F    15
NO 17    7D10    77
NO 18    7D11    23
NO 19    7D12    0D
NO 20    7D13    20
NO 21    7D14    FB
NO 22    7D15    DD
NO 23    7D16    7E
NO 24    7D17    00
NO 25    7D18    62
NO 26    7D19    6B
NO 27    7D1A    D6
NO 28    7D1B    05
NO 29    7D1C    38
NO 30    7D1D    03
NO 31    7D1E    23
NO 32    7D1F    18
NO 33    7D20    F9
NO 34    7D21    34
NO 35    7D22    DD
NO 36    7D23    23
NO 37    7D24    10
NO 38    7D25    EF
NO 39    7D26    C9
NO 40    7D27    05
NO 41    7D28    64
NO 42    7D29    2D
NO 43    7D2A    00
NO 44    7D2B    49
NO 45    7D2C    15
NO 46    7D2D    00
NO 47    7D2E    01
NO 48    7D2F    02
NO 49    7D30    03
NO 50    7D31    04
NO 51    7D32    05
NO 52    7D33    06
NO 53    7D34    07
NO 54    7D35    08
NO 55    7D36    09
NO 56    7D37    10
NO 57    7D38    11
NO 58    7D39    12
NO 59    7D3A    13
NO 60    7D3B    14
NO 61    7D3C    15
NO 62    7D3D    16
NO 63    7D3E    17
NO 64    7D3F    18
NO 65    7D40    19
NO 66    7D41    20
---------------------------------------
```

《第14－27図・その２》 第14－26図の機械語を一つ一つメモリに入れ実行前後の内容を比較したもの

```
    MACRO-80 3.31   03-JUL-79        PAGE    1-1
7D36    09                    04400          DB       9H
7D37    10                    04450          DB       10H
7D38    11                    04500          DB       11H
7D39    12                    04600          DB       12H
7D3A    13                    04700          DB       13H
7D3B    14                    04800          DB       14H
7D3C    15                    04900          DB       15H
7D3D    16                    05000          DB       16H
7D3E    17                    05100          DB       17H
7D3F    18                    05200          DB       18H
7D40    19                    05300          DB       19H
7D41    20                    05400          DB       20H
                              05500          END

     MACRO-80 3.31   03-JUL-79        PAGE    S
MACROS.
SYMBOLS.
DATA    7D28    JMP     7D21    LOOP1   7D10    LOOP2   7D15
LOOP3   7D1A    SIZE    7D27    TABLE   7D2D
NO  FATAL ERROR(S)
```

```
** AFFTER RUN
------------------------------------
 NO 1          7D00          11
 NO 2          7D01          2D
 NO 3          7D02          7D
 NO 4          7D03          DD
 NO 5          7D04          21
 NO 6          7D05          28
 NO 7          7D06          7D
 NO 8          7D07          3A
 NO 9          7D08          27
 NO 10         7D09          7D
 NO 11         7D0A          47
 NO 12         7D0B          62
 NO 13         7D0C          6B
 NO 14         7D0D          97
 NO 15         7D0E          0E
 NO 16         7D0F          15
 NO 17         7D10          77
 NO 18         7D11          23
 NO 19         7D12          0D
 NO 20         7D13          20
 NO 21         7D14          FB
 NO 22         7D15          DD
 NO 23         7D16          7E
 NO 24         7D17          00
 NO 25         7D18          62
 NO 26         7D19          6B
 NO 27         7D1A          D6
 NO 28         7D1B          05
 NO 29         7D1C          38
 NO 30         7D1D          03
 NO 31         7D1E          23
 NO 32         7D1F          18
 NO 33         7D20          F9
 NO 34         7D21          34
 NO 35         7D22          DD
 NO 36         7D23          23
 NO 37         7D24          10
 NO 38         7D25          EF
 NO 39         7D26          C9
 NO 40         7D27          05
 NO 41         7D28          64
 NO 42         7D29          2D
 NO 43         7D2A          00
 NO 44         7D2B          49
 NO 45         7D2C          15
 NO 46         7D2D          01
 NO 47         7D2E          00
 NO 48         7D2F          00
 NO 49         7D30          00
 NO 50         7D31          01
 NO 51         7D32          00
 NO 52         7D33          00
 NO 53         7D34          00
 NO 54         7D35          00
 NO 55         7D36          01
 NO 56         7D37          00
 NO 57         7D38          00
 NO 58         7D39          00
 NO 59         7D3A          00
 NO 60         7D3B          01
 NO 61         7D3C          00
 NO 62         7D3D          00
 NO 63         7D3E          00
 NO 64         7D3F          00
 NO 65         7D40          00
 NO 66         7D41          01
------------------------------------
```

データは，10進数になおせば100，45，0，73，21の五つとしました。また，実行後には度数の入るM(TABLE）以下21個のメモリには0から２０までの数を入れておきました。クリアされているかどうか，確認をするためです。

データは，0

0～4………一つ

20～24………一つ

45～49………一つ

7ϕ～74………一つ

100以上 ……一つ

ですから，M(TABLE）以下の21個のメモリの内容は，

1-0-0-0-1-0-0-0-0-1-0-0-0-0-1-0-0-0-0-0-1 となるはずです。

これを確かめるために，**第14－26図**から得られた機械語を，７D00番地以降のデータに入れ実行してみたものが**第14－27図**です。これより，プログラムが正常に働いていることが確認できます。

以上で，プログラム〔E〕は完成です。

〔F〕 合計と平均を求めるプログラム

連続するメモリに入っている２進数のデータの合計(ただし２ワードを越えない)と，平均を求めるプログラムです。平均は，小数点以下第３位を四捨五入したときの，第２位までを求めるようにしてあります。

全データは，１ワードに納まっているもの，としてありますから，平均の整数部も１ワードですみます。したがって，平均には２ワードを用意すれば，小数点以下第２位までを納めるのに十分です。

さて，アルゴリズムですが，合計のそれはたいしたことはありません。データはすべて，１ワードに納まるものですが合計を計算するときには，２ワードどうしの足し算の命令を使うのが得策です。たとえば，Dにゼロを入れ，Eに個々のデータを送り込み，

HL←HL＋DE

命令を使って，HLに和を求めていけばよいでしょう。

平均を求めるためには，割り算のサブルーチンが必要になります。小数点以下を計算するのであれば，さらに掛け算のサブルーチンも必要になります。

合計をデータ数で割ったときの商は，平均の整数部になります。そのときの余りを100倍して，それをデータ数で割ったときの商が小数第１位，第２位の値になります。四捨五入は，そのときの余りを２倍して，データ数と比較します。もし，データ数の方が大きければ，前の商はそのままです。同じか，余りの２倍の方が大きい場合には，商に１を加わえます。これで，小数第３位を四捨五入したことになります。

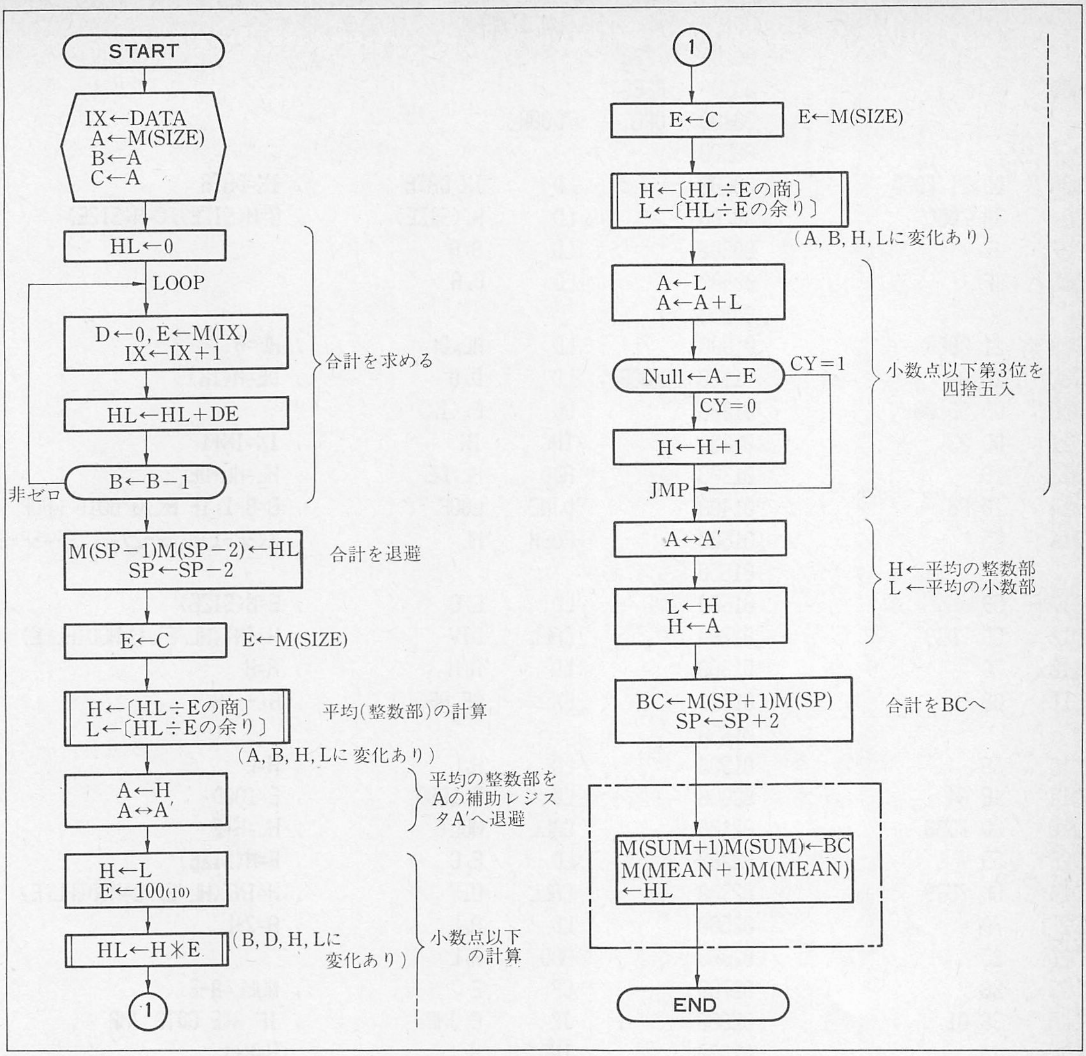
さて，以上のことを注意して，**第14－28図**のようなフローチャートを書いてみました。

２ワードにまたがる合計は，BCレジスタペアに，平均の整数部はHレジスタに，小数点以下はLレジスタに入るようになっています。

後で，独立したサブルーチンとしても使えるように，外部メモリのアドレスが必要な部分（データ数の入っているメモリのアドレスと，データの先頭アドレス）は最前部にまとめてあります。そのために，レジスタCが活用されています。

また，二つのサブルーチンを用いるわけですが，サブルーチン内で使われているレジスタは，当然サブルーチンを実行すれば，その値を変えてしまいます。したがってサブルーチンの前後で，データの受け渡しはできませんので注意が必要です。**第14－28図**の中でも，そのためにレジスタAの補助レジスタに，平均の整数部を退避させています。

一般に，２ワードにまたがるデータの退避は，HLレジスタペア，DEレジスタペアの入れ換えで可能な場合はこれを使います。そうしない場合は，PUSH，POP命令で行なうのが効率的です。１ワードのデータの場合，他に空いているレジスタがなければ，レジスタAの補助レジスタを用いるのがよいでしょう。ただし，この場合，各フラグの状態も入れ代わってしまいますから注意が必要です。



《第14−28図》 合計と平均を求めるプログラムのためのフローチャート

第14−28図のフローチャートをもとに，アセンブラプログラミングしたものが**第14−29図**です。割り算，掛け算のサブルーチンは前にあげたものと同じものです。

マイクロソフト社のアセンブラにより，リストの見方の注意点については，何度もお話ししてきましたが，16進数が切れ目なく四つつながって表示されているところには注意してください。

データは，10進数になおせば，99，98，97，96，95の五つとしてあります。この合計は485，平均は97.00ですから，プログラムが正常であれば，

M(SUM+1)M(SUM)=01E5(16)

M(MEAN+1)M(MEAN)=6100(16)

となるはずです。もう少し具体的に書けば

M(7D62)=E5

M(7D63)=01

M(7D64)=00

M(7D65)=61

となるはずです。

これを確かめるために，**第14−29図**から得られた機械語を一つ一つメモリに書き込み，実際にこのプログラムを動かしてみて，メモリの内容がどのように変化したかを見たものが，**第14−30図**です。これより，このプログラムが正常に働いていることが確認できます。

しかし，このデータの場合，平均の小数点以下の確認ができません。そこでデータを10進数になおせば，100，110，0となるものにして，平均が66

```
      MACRO-80 3.31   03-JUL-79        PAGE    1
                           00100  ;        SUM & MEAN
                           00200  .Z80
 0000'                     00300  ASEG
                           00400  ORG      7D00H
                           00500  ;
 7D00    DD 21 7D5D        00600           LD      IX,DATA         ; IX=DATA
 7D04    3A 7D5C           00700           LD      A,(SIZE)        ; B=M(SIZE),C=M(SIZE)
 7D07    47                00800           LD      B,A
 7D08    4F                00900           LD      C,A
                           00950  ;
 7D09    21 0000           01000           LD      HL,0            ; HL=0
 7D0C    16 00             01100  LOOP:    LD      D,0             ; DE=M(IX)
 7D0E    DD 5E 00          01200           LD      E,(IX)
 7D11    DD 23             01250           INC     IX              ; IX=IX+1
 7D13    19                01300           ADD     HL,DE           ; HL=HL+DE
 7D14    10 F6             01400           DJNZ    LOOP            ; B=B-1,IF B<>0 GOTO LOOP
 7D16    E5                01500           PUSH    HL              ; M(SP-1)M(SP-2)=HL,SP=SP-2
                           01550  ;
 7D17    59                01600           LD      E,C             ; E=M(SIZE)
 7D18    CD 7D39           01700           CALL    DIV             ; H=INT(HL/E),L=MOD(HL,E)
 7D1B    7C                01800           LD      A,H             ; A=H
 7D1C    08                01820           EX      AF,AF'          ; A<-->A'
                           01850  ;
 7D1D    65                01900           LD      H,L             ; H=L
 7D1E    1E 64             02000           LD      E,100D          ; E=100D
 7D20    CD 7D50           02100           CALL    MULT            ; HL=H*E
 7D23    59                02200           LD      E,C             ; E=M(SIZE)
 7D24    CD 7D39           02300           CALL    DIV             ; H=INT(HL/E),L=MOD(HL,E)
 7D27    7D                02500           LD      A,L             ; A=2*L
 7D28    85                02600           ADD     A,L
 7D29    BB                02700           CP      E               ; NULL=A-E
 7D2A    38 01             02800           JR      C,JMP           ; IF A<E GOTO JMP
 7D2C    24                02900           INC     H               ; H=H+1
 7D2D    08                03000  JMP:     EX      AF,AF'          ; A<-->A'
 7D2E    6C                03100           LD      L,H             ; L=H
 7D2F    67                03200           LD      H,A             ; H=A
 7D30    C1                03300           POP     BC              ; BC=M(SP+1)M(SP),SP=SP+2
                           03400  ;
 7D31    ED 43 7D62        03500           LD      (SUM),BC        ; M(SUM+1)M(SUM)=BC
 7D35    22 7D64           03600           LD      (MEAN),HL       ; M(MEAN+1)M(MEAN)=HL
 7D38    C9                03700           RET                     ; END
                           03800  ;
                           03900  ;------ SUBROUTINE ( H=INT(HL/E), L=MOD(HL,E) ) -------
                           04000  ;                A,B,H,L : AFFECTED
                           04100  ;                C,D,E   ; NOT AFFECTED
                           04200  ;
```

《第14－29図》 合計と平均を求めるプログラム

```
     MACRO-80 3.31   03-JUL-79        PAGE    1-1
7D39    06 08          DIV.    LD      B,8H            , B=8
7D3B    AF                     XOR     A               , CY=0
7D3C    ED 6A          DLOOP.  ADC     HL,HL           , HL=HL+HL+CY
7D3E    7C                     LD      A,H             , A=H
7D3F    38 03                  JR      C,DJMP1         , IF CY=1 GOTO DJMP1
7D41    BB                     CP      E               , NULL=A-E
7D42    38 03                  JR      C,DJMP2         , IF CY=1 GOTO DJMP2
7D44    93             DJMP1.  SUB     E               , A=A-E
7D45    67                     LD      H,A             , H=A
7D46    AF                     XOR     A               , CY=0
7D47    3F             DJMP2.  CCF                     , IF CY=0 THEN CY=1 ELSE CY=0
7D48    10 F2                  DJNZ    DLOOP           , B=B-1,IF B<>0 GOTO DLOOP
7D4A    CB 15                  RL      L               , L=2*L+CY
7D4C    7C                     LD      A,H             , H<-->L
7D4D    65                     LD      H,L
7D4E    6F                     LD      L,A
7D4F    C9                     RET                     , RETURN
                       ,----------------------------------------------------
                       ,
                       ,------ SUBROUTINE ( HL=H*L ) ------------------------
                       ,               B,D,H,L . AFFECTED
                       ,               A,C,E   . NOT AFFECTED
                       ,
7D50    16 00          MULT.   LD      D,0             , D=0
7D52    6A                     LD      L,D             , L=0
7D53    06 08                  LD      B,8H            , B=8
7D55    29             MLOOP.  ADD     HL,HL           , HL=2*HL
7D56    30 01                  JR      NC,MJMP         , IF CY=0 GOTO MJMP
7D58    19                     ADD     HL,DE           , HL=HL+DE
7D59    10 FA          MJMP.   DJNZ    MLOOP           , B=B-1,IF B<>0 GOTO MLOOP
7D5B    C9                     RET                     , RETURN
                       ,----------------------------------------------------
                       ,
7D5C    05             SIZE.   DB      5D              , M(SIZE)=5
7D5D    63             DATA.   DB      99D
7D5E    62                     DB      98D
7D5F    61                     DB      97D
7D60    60                     DB      96D
7D61    5F                     DB      95D
7D62    00             SUM.    DB      0
7D63    00                     DB      0
7D64    00             MEAN.   DB      0
7D65    00                     DB      0
                               END
      MACRO-80 3.31   03-JUL-79       PAGE    S
MACROS.
SYMBOLS.
DATA    7D5D    DIV     7D39    DJMP1   7D44    DJMP2   7D47
DLOOP   7D3C    JMP     7D2D    LOOP    7D0C    MEAN    7D64
MJMP    7D59    MLOOP   7D55    MULT    7D50    SIZE    7D5C
SUM     7D62
NO  FATAL ERROR(S)
```

** BEFORE RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	5D
NO 4	7D03	7D
NO 5	7D04	3A
NO 6	7D05	5C
NO 7	7D06	7D
NO 8	7D07	47
NO 9	7D08	4F
NO 10	7D09	21
NO 11	7D0A	00
NO 12	7D0B	00
NO 13	7D0C	16
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	5E
NO 17	7D10	00
NO 18	7D11	DD
NO 19	7D12	23
NO 20	7D13	19
NO 21	7D14	10
NO 22	7D15	F6
NO 23	7D16	E5
NO 24	7D17	59
NO 25	7D18	CD
NO 26	7D19	39
NO 27	7D1A	7D
NO 28	7D1B	7C
NO 29	7D1C	08
NO 30	7D1D	65
NO 31	7D1E	1E
NO 32	7D1F	64
NO 33	7D20	CD
NO 34	7D21	50
NO 35	7D22	7D
NO 36	7D23	59
NO 37	7D24	CD
NO 38	7D25	39
NO 39	7D26	7D
NO 40	7D27	7D
NO 41	7D28	85
NO 42	7D29	BB
NO 43	7D2A	38
NO 44	7D2B	01
NO 45	7D2C	24
NO 46	7D2D	08
NO 47	7D2E	6C
NO 48	7D2F	67
NO 49	7D30	C1
NO 50	7D31	ED

NO 51	7D32	43
NO 52	7D33	62
NO 53	7D34	7D
NO 54	7D35	22
NO 55	7D36	64
NO 56	7D37	7D
NO 57	7D38	C9
NO 58	7D39	06
NO 59	7D3A	08
NO 60	7D3B	AF
NO 61	7D3C	ED
NO 62	7D3D	6A
NO 63	7D3E	7C
NO 64	7D3F	38
NO 65	7D40	03
NO 66	7D41	BB
NO 67	7D42	38
NO 68	7D43	03
NO 69	7D44	93
NO 70	7D45	67
NO 71	7D46	AF
NO 72	7D47	3F
NO 73	7D48	10
NO 74	7D49	F2
NO 75	7D4A	CB
NO 76	7D4B	15
NO 77	7D4C	7C
NO 78	7D4D	65
NO 79	7D4E	6F
NO 80	7D4F	C9
NO 81	7D50	16
NO 82	7D51	00
NO 83	7D52	6A
NO 84	7D53	06
NO 85	7D54	08
NO 86	7D55	29
NO 87	7D56	30
NO 88	7D57	01
NO 89	7D58	19
NO 90	7D59	10
NO 91	7D5A	FA
NO 92	7D5B	C9
NO 93	7D5C	05
NO 94	7D5D	63
NO 95	7D5E	62
NO 96	7D5F	61
NO 97	7D60	60
NO 98	7D61	5F
NO 99	7D62	00
NO 100	7D63	00
NO 101	7D64	00
NO 102	7D65	00

《第14－30図》 第14－29図の機械語を一つ一つメモリに入れ実行前後の内容を比較したもの

** AFFTER RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	5D
NO 4	7D03	7D
NO 5	7D04	3A
NO 6	7D05	5C
NO 7	7D06	7D
NO 8	7D07	47
NO 9	7D08	4F
NO 10	7D09	21
NO 11	7D0A	00
NO 12	7D0B	00
NO 13	7D0C	16
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	5E
NO 17	7D10	00
NO 18	7D11	DD
NO 19	7D12	23
NO 20	7D13	19
NO 21	7D14	10
NO 22	7D15	F6
NO 23	7D16	E5
NO 24	7D17	59
NO 25	7D18	CD
NO 26	7D19	39
NO 27	7D1A	7D
NO 28	7D1B	7C
NO 29	7D1C	08
NO 30	7D1D	65
NO 31	7D1E	1E
NO 32	7D1F	64
NO 33	7D20	CD
NO 34	7D21	50
NO 35	7D22	7D
NO 36	7D23	59
NO 37	7D24	CD
NO 38	7D25	39
NO 39	7D26	7D
NO 40	7D27	7D
NO 41	7D28	85
NO 42	7D29	BB
NO 43	7D2A	38
NO 44	7D2B	01
NO 45	7D2C	24
NO 46	7D2D	08
NO 47	7D2E	6C
NO 48	7D2F	67
NO 49	7D30	C1
NO 50	7D31	ED
NO 51	7D32	43
NO 52	7D33	62
NO 53	7D34	7D
NO 54	7D35	22
NO 55	7D36	64
NO 56	7D37	7D
NO 57	7D38	C9
NO 58	7D39	06
NO 59	7D3A	08
NO 60	7D3B	AF
NO 61	7D3C	ED
NO 62	7D3D	6A
NO 63	7D3E	7C
NO 64	7D3F	38
NO 65	7D40	03
NO 66	7D41	BB
NO 67	7D42	38
NO 68	7D43	03
NO 69	7D44	93
NO 70	7D45	67
NO 71	7D46	AF
NO 72	7D47	3F
NO 73	7D48	10
NO 74	7D49	F2
NO 75	7D4A	CB
NO 76	7D4B	15
NO 77	7D4C	7C
NO 78	7D4D	65
NO 79	7D4E	6F
NO 80	7D4F	C9
NO 81	7D50	16
NO 82	7D51	00
NO 83	7D52	6A
NO 84	7D53	06
NO 85	7D54	08
NO 86	7D55	29
NO 87	7D56	30
NO 88	7D57	01
NO 89	7D58	19
NO 90	7D59	10
NO 91	7D5A	FA
NO 92	7D5B	C9
NO 93	7D5C	05
NO 94	7D5D	63
NO 95	7D5E	62
NO 96	7D5F	61
NO 97	7D60	60
NO 98	7D61	5F
NO 99	7D62	E5
NO 100	7D63	01
NO 101	7D64	00
NO 102	7D65	61

** BEFORE RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	5D
NO 4	7D03	7D
NO 5	7D04	3A
NO 6	7D05	5C
NO 7	7D06	7D
NO 8	7D07	47
NO 9	7D08	4F
NO 10	7D09	21
NO 11	7D0A	00
NO 12	7D0B	00
NO 13	7D0C	16
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	5E
NO 17	7D10	00
NO 18	7D11	DD
NO 19	7D12	23
NO 20	7D13	19
NO 21	7D14	10
NO 22	7D15	F6
NO 23	7D16	E5
NO 24	7D17	59
NO 25	7D18	CD
NO 26	7D19	39
NO 27	7D1A	7D
NO 28	7D1B	7C
NO 29	7D1C	08
NO 30	7D1D	65
NO 31	7D1E	1E
NO 32	7D1F	64
NO 33	7D20	CD
NO 34	7D21	50
NO 35	7D22	7D
NO 36	7D23	59
NO 37	7D24	CD
NO 38	7D25	39
NO 39	7D26	7D
NO 40	7D27	7D
NO 41	7D28	85
NO 42	7D29	BB
NO 43	7D2A	38
NO 44	7D2B	01
NO 45	7D2C	24
NO 46	7D2D	08
NO 47	7D2E	6C
NO 48	7D2F	67
NO 49	7D30	C1
NO 50	7D31	ED
NO 51	7D32	43
NO 52	7D33	62
NO 53	7D34	7D
NO 54	7D35	22
NO 55	7D36	64
NO 56	7D37	7D
NO 57	7D38	C9
NO 58	7D39	06
NO 59	7D3A	08
NO 60	7D3B	AF
NO 61	7D3C	ED
NO 62	7D3D	6A
NO 63	7D3E	7C
NO 64	7D3F	38
NO 65	7D40	03
NO 66	7D41	BB
NO 67	7D42	38
NO 68	7D43	03
NO 69	7D44	93
NO 70	7D45	67
NO 71	7D46	AF
NO 72	7D47	3F
NO 73	7D48	10
NO 74	7D49	F2
NO 75	7D4A	CB
NO 76	7D4B	15
NO 77	7D4C	7C
NO 78	7D4D	65
NO 79	7D4E	6F
NO 80	7D4F	C9
NO 81	7D50	16
NO 82	7D51	00
NO 83	7D52	6A
NO 84	7D53	06
NO 85	7D54	08
NO 86	7D55	29
NO 87	7D56	30
NO 88	7D57	01
NO 89	7D58	19
NO 90	7D59	10
NO 91	7D5A	FA
NO 92	7D5B	C9
NO 93	7D5C	03
NO 94	7D5D	64
NO 95	7D5E	64
NO 96	7D5F	00
NO 97	7D60	00
NO 98	7D61	00
NO 99	7D62	00
NO 100	7D63	00
NO 101	7D64	00
NO 102	7D65	00

《第14−31図》 小数点以下の確認のためにデータを100，100，0の三つにしたもの

** AFFTER RUN

NO 1	7D00	DD
NO 2	7D01	21
NO 3	7D02	5D
NO 4	7D03	7D
NO 5	7D04	3A
NO 6	7D05	5C
NO 7	7D06	7D
NO 8	7D07	47
NO 9	7D08	4F
NO 10	7D09	21
NO 11	7D0A	00
NO 12	7D0B	00
NO 13	7D0C	16
NO 14	7D0D	00
NO 15	7D0E	DD
NO 16	7D0F	5E
NO 17	7D10	00
NO 18	7D11	DD
NO 19	7D12	23
NO 20	7D13	19
NO 21	7D14	10
NO 22	7D15	F6
NO 23	7D16	E5
NO 24	7D17	59
NO 25	7D18	CD
NO 26	7D19	39
NO 27	7D1A	7D
NO 28	7D1B	7C
NO 29	7D1C	08
NO 30	7D1D	65
NO 31	7D1E	1E
NO 32	7D1F	64
NO 33	7D20	CD
NO 34	7D21	50
NO 35	7D22	7D
NO 36	7D23	59
NO 37	7D24	CD
NO 38	7D25	39
NO 39	7D26	7D
NO 40	7D27	7D
NO 41	7D28	85
NO 42	7D29	BB
NO 43	7D2A	38
NO 44	7D2B	01
NO 45	7D2C	24
NO 46	7D2D	08
NO 47	7D2E	6C
NO 48	7D2F	67
NO 49	7D30	C1
NO 50	7D31	ED
NO 51	7D32	43
NO 52	7D33	62
NO 53	7D34	7D
NO 54	7D35	22
NO 55	7D36	64
NO 56	7D37	7D
NO 57	7D38	C9
NO 58	7D39	06
NO 59	7D3A	08
NO 60	7D3B	AF
NO 61	7D3C	ED
NO 62	7D3D	6A
NO 63	7D3E	7C
NO 64	7D3F	38
NO 65	7D40	03
NO 66	7D41	BB
NO 67	7D42	38
NO 68	7D43	03
NO 69	7D44	93
NO 70	7D45	67
NO 71	7D46	AF
NO 72	7D47	3F
NO 73	7D48	10
NO 74	7D49	F2
NO 75	7D4A	CB
NO 76	7D4B	15
NO 77	7D4C	7C
NO 78	7D4D	65
NO 79	7D4E	6F
NO 80	7D4F	C9
NO 81	7D50	16
NO 82	7D51	00
NO 83	7D52	6A
NO 84	7D53	06
NO 85	7D54	08
NO 86	7D55	29
NO 87	7D56	30
NO 88	7D57	01
NO 89	7D58	19
NO 90	7D59	10
NO 91	7D5A	FA
NO 92	7D5B	C9
NO 93	7D5C	03
NO 94	7D5D	64
NO 95	7D5E	64
NO 96	7D5F	00
NO 97	7D60	00
NO 98	7D61	00
NO 99	7D62	C8
NO 100	7D63	00
NO 101	7D64	43
NO 102	7D65	42

67になるがどうか確認してみることにしましょう。

第30図の1093以降のデータを，

3，64，64，00，00，00，00，00，00，00　にしてから，実行すればよいわけです。合計は10進数になおせば2　ですから，10進数ではＣ８，平均の整数部は10進数になおせば66ですから，16進数では42，小数点以下は67ですから，16進数では43になるはずです。

これを確かめてみたのが**第14－31図**です。上で述べた通りの結果が得られていることが，ここで確認できます。

以上で，プログラム〔Ｆ〕は完成です。

簡単システムを想定して，〔Ａ〕から〔Ｆ〕のプログラムを作ってきました。これらのプログラムをサブルーチン化して，簡単なメインプログラムのもとで働かせるようにすれば，想定したシステムはできあがります（メインの作り方によっては，プログラム〔Ｃ〕は必要なかったかもしれません）。データ数を入れておくメモリのアドレス，データの先頭アドレス，各結果を出力する場合の先頭アドレス等のシンボルに注意して，また手持のメモリ数に注意して，メインプログラムを作ってみてください。

15 固定・浮動小数点演算パッケージ

はじめに

Z80は，数多くの命令セットを持ち，また8080などに比べて高速であることなど，高いポテンシャルを秘めたプロッサであると言うことができます。

けれども，Z80，を数値演算などの目的に使用しようとする時には，他の8ビットマイクロプロセッサと同様に，乗除算や関数の命令を有していないため，機械語やアセンブラでプログラムを作成する場合には多大の労力を必要とします。またBASICなどの言語を使用したプログラムでは，処理速度が遅かったり，あるいはシステムが必要以上に大きくなったりする場合があります。

こういった問題を解決するためには，アセンブラ言語による演算サブルーチンをいくつか用意して，必要に応じてそれを使えばよいという事になります。幸いにして8080用のアセンブラで書かれた演算サブルーチンは，雑誌などを通じて数多く発表されておりますので，Z80を使用したマイコンシステムをお持ちの方も，そういった演算サブルーチンを使用することができます。

しかしながら，Z80と8080との機械語レベルでのコンパチビリティーに頼って8080用に書かれたプログラムばかり使用していたのでは，Z80を使用する妙味が半減してしまいます（これは演算サブルーチンに限った事ではありませんが)。

そこで，本章では，Z80の特徴を，十分に生かした演算サブルーチンを紹介します。

紹介するプログラムは，固定小数点数ならびに浮動小数点数の四則演算，および三角関数や平方根などの，いくつかの演算サブルーチンより成り立っています。

プログラムに関する詳細な説明については後述致しますが，プログラムはサブルーチン形式になっておりますので，例えばこのプログラムと適当な入出力プログラムを用意して，電卓と同じように使用されても結構ですし，また必要に応じてプログラムの一部を利用されても結構です。

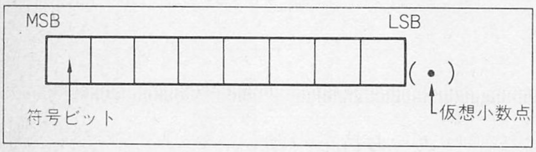
このプログラムでとり扱う数値は，すべて通常の2進数であり，関数演算についてはその精度を固定しておりますが，四則演算については任意の

精度で演算が行なえるようになっておりますので，電卓の代用ばかりでなく，様々な用途に利用できる事と思います。私としては皆さんのソフトウェアライブラリーの一つに加えていただき，多くの場所でこのプログラムが使用される事を期待しております。

さて，プログラムについて解説する前に，演算ルーチンのアルゴリズムなどについて，簡単に述べておきましょう。

15－1 固定小数点数の演算

固定小数点数といっても，マイクロコンピュータには小数点を表わすための特別なレジスタやメモリなどがあるわけではありませんので，15－1図のように取り扱うデータの最下位ビットの右側に小数点があるものと考えて，データを扱います。



《第15－1図》固定小数点数の構造

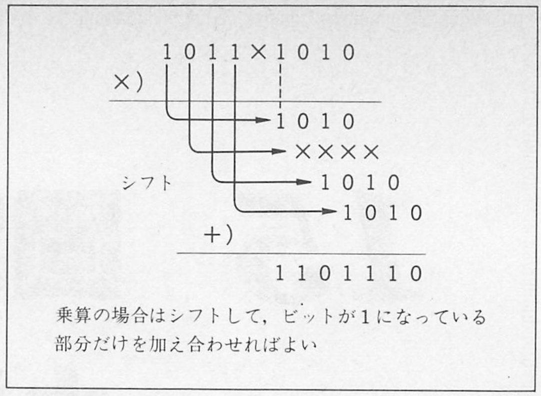
15－1－1 乗算・除算のアルゴリズム

データ長が第15－1図のように8ビットになっている場合には，加算と減算については，Z80の持っている命令でそのまま演算を行なうことができます。

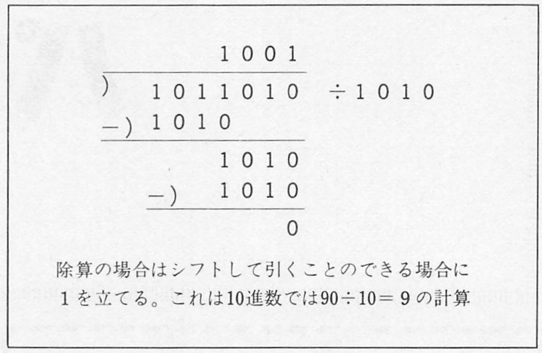
乗算と除算については加算あるいは減算をくり返すことによっても行なうことができますが，この方法では，データの大きさによって演算に必要とする時間が大幅に変化しますし，効率のよい方法とは言えません。

そこで一般に，マイコンで乗算を行なう場合には，第15－2図のように，シフトしながら数値を加え合わせていく方法で乗除算を行ないます。この方法は，一見すると特別な方法のように感じられるかもしれませんが，私達が日常10進数の筆算でよく使っている方法なのです。

除算についても乗算と本質的には同じで，この場合には，第15－3図のようにシフトしながら，数値を差し引いていくアルゴリズムによって演算



《第15－2図》10進数の11（＝1011)2掛ける10進数の10（＝1010)2の計算
結果は110（＝1101110)2



《第15－3図》

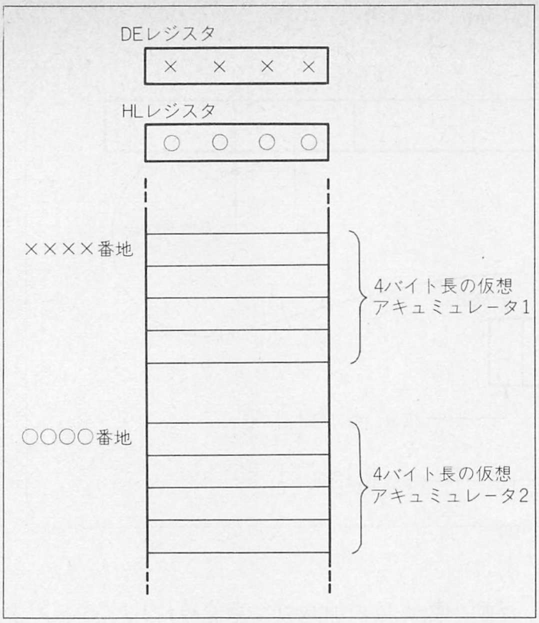
を行ないます。

15－1－2 多精度を扱う方法

言うまでもなくZ80は8ビットのマイクロプロセッサですが、この8ビットというデータ長で表わすことのできる数値は符号付きの整数では、−128から＋127にすぎません。けれども実際の数値計算で問題とするデータはさらに大きな数値であったり，あるいは，浮動小数点数の場合には，多くの有効桁数を持っていたりして，8ビットにおさまらない場合が多数くあります。

多精度のデータをとり抜う方法としてはいくつかあり，例えば16ビット長のデータに対しては，Z80の場合には，DEレジスタ，およびHLレジスタを用いて，データを扱うこともできます。この場合Z80の裏レジスタをうまく活用すれば，速度の速いプログラムを作成することが可能となります。

今回のプログラムでは，より一般的な方法として，第15－4図のようにメモリを仮想のアキュム



《第15－4図》メモリを仮想アキュミュレータとして使う

レータと考えてデータを扱います。

この方法を用いれば，データの精度は，実際上無制限に増加することができます。前述しましたように四則演算を行なううえでの基本となるのは，加算と減算ですが，他精度の加減算を行なう場合には，キャリーフラグを，有効に利用しなくてはなりません。この場合に，ADCやSBCなどのキャリーを含んだ加減算命令は，非常に有効なものとなります。（16ビットのマイクロプロセッサの中には，このようなキャリーを含んだ加減算命令のないものもあります。この場合，多精度の演算を行なううえでどのような不都合が起こるか，考えてみて下さい。

15－2浮動小数点数の演算

数値計算を行なう場合には前述の固定小数点形式のほかに，浮動小数点形式で数値を表現できると便利です。

浮動小数点の形式で，私たちが日常みなれているものは10のべき乗形式でしょう。例えば

$3,1415 \times 10^{8}$

$1,234 \times 10^{-6}$

のように，10進数で表現された仮数部と指数部から成り立っています。計算機で，浮動小数点数を表現する場合には，２のべき乗，８のべき乗，16のべき乗などの方法がありますが，ここでは16のべき乗による表現を用いています。

15－２－１浮動小数点数の構造

16のべき乗を用いる理由としては，とり扱うことのできる数値の範囲が広いことがあげられます。

先ほどの3.14×10^{3}という数値について考えてみましょう。四捨五入の結果この数値が得られたものとすると，この数値の表わしているデータの大きさ（xとします）は，

$$3.135 \times 10^{3} \leqq x < 3.145 \times 10^{3}$$

という幅を持っていると考えられます。

つまり，浮動小数点数でデータを表わす場合には，その最下位桁の大きさ（この場合は0.01×10^{3}）という誤差を持っているものと考えなくてはなりません。

この誤差は，データの表現が２のべき乗の場合は，2^{n}の形ですので小さなものですが，データの表現が16のべき乗の場合は16^{n}の形となり，くり返して計算を行なううちに大きな誤差を発生する可能性があります。この点については２のべき乗による表現の方がすぐれています。

けれども，同一のビット数で表現することのできる数値の範囲は，２のべき乗の形式に比べて，16のべき乗の形式の方がはるかに大きく，それぞれ一長一短があります。

ここでは16のべき乗による表現を使用します。

この表現によって，例えば10進数の0.1を表わしますと，

$$0.1 = 1 \times 16^{-1} + 9 \times 16^{-2} + 9 \times 16^{-4} + \cdots$$

のように無限級数になります。

実際の構造は**第15－5図**のように，仮数部７バイト，および指数部１バイト（符号ビットを含む）から構成されています。

この構造によって表現することのできる数値は

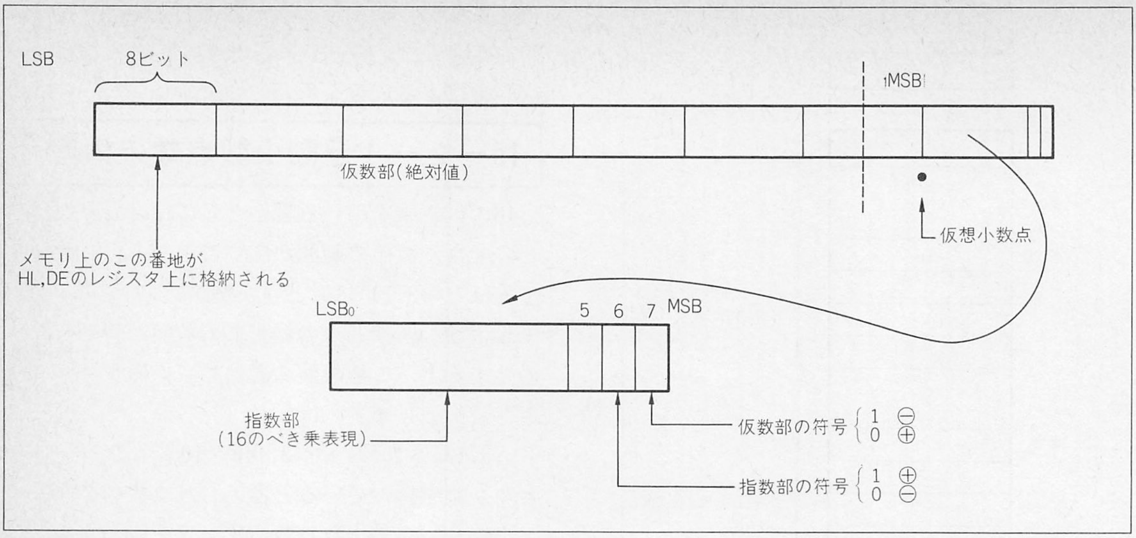
$$0.0625 \times 16^{-64} \leqq |x| < 1 \times 16^{63}$$

10進法では，およそ

$$5.397 \times 10^{-79} \leqq |x| < 7.237 \times 10^{75}$$

となっています。

浮動小数点数によって数値を現わす場合に注意しておかなくてはならないことに，正規化という問題があります。これは計算機により演算を行な



《第15－5図》浮動小数点数の構造

う場合に限りませんが，ほとんど同じ大きさの数の減算を行なった場合などに，桁落ちといって有効桁数が減少するのを防ぐ処理です。

例えば，

$$0.333\times10^3-0.332\times10^3=0.001\times10^3$$

のような演算を行なう場合には，これをそのまま計算機に行なわせたのでは，最初３桁あった有効桁数が，結果としては１桁になってしまいます。これを防ぐ為に，正規化という処理を行ないます。これは，

$$0.001\times10^3=0.100\times10^1$$

のように，常に，ある有効桁数を確保する処理で，す。

ここで紹介する演算ルーチンでは

$$0.mmnnn\times16^n$$

の形で正規化を行ない有効桁数の減少を防いでいます。

15－２－２浮動小数点数の乗除算

浮動小数点数の乗除算も，固定小数点数の場合と基本的には同様です。まず，仮数部について固定小数点数と同様な演算を行なった後に，得られた結果に対して，上に述べた正規化を行ないます。指数部については，乗算の場合には加算，除算の場合には減算となりますが，これに正規化を行なった場合のべき乗数を補正して最終結果とします。

15－２－３浮動小数点数の加減算

浮動小数点数の加減算は乗除算に比べると，すこし手間がかかります。その手順はまず，加減算しようとする数値（いずれも正規化されているものとします）の両方の指数部を比較して，両者が同一の指数部を持つ様に，仮数部を調整します。この後で，仮数部に対して加減算を行ない，さらに得られた結果に対して正規化を行なわなくてはなりません。

15－２－４丸めについて

これも計算機で演算を行なう場合に限ったことではありませんが，ここでは10進数演算の場合の四捨五入に相当する七捨八入の処理を行なっています。もちろん，この処理を行なう為には，必要とする桁数よりもさらに１桁以上多くの桁について，演算を行なう必要があります。

15－２－５演算ルーチンの文法

演算ルーチンのプログラム開発は，マイクロソフト社のマイクロ80の下で行ないました。

プログラムを皆さんのマイコンシステムに移植する際に必要なプログラム・リストの見方などに関する２，３の注意点を挙げておきましょう。

15－３アセンブラについて

アセンブラは，Ｚ80のニーモニックで書かれて

おり，特に変わったものではありません。プログラム・リストは，左から行番号，アドレス，機械語，ラベル，ニーモニックオペランドの順に書かれています。但し，小さなことですが，相対ジャンプ命令のオペランド欄の書き方で，－$が省略されていますので注意して下さい。

また，アドレスはすべて0番地から割り振ってありますので，プログラムを各自のマイコンシステムに機械語レベルで移植する場合には，注意してアドレス計算を行なって下さい。

アドレスの書き換えが必要な機械語には，4桁の16進数の後にアポストロフィーが振ってあります。アスタリシック(＊)が振ってある4桁数値は，外部のプログラムのアドレスですから，これについてもやはり，アドレスの変更が必要です。アドレスの変更には16進計算尺があるととても便利です。

尚，このアドレスは上位，下位の順に書かれていますが，実際にマイコンにプログラムを移植する場合には，この逆にメモリ上に書き込む必要があります。つまり，026C’と書かれている場合，実際には6C，02の順に移植を行ないます。

何の添え記号も振っていない4桁数値は，データなどの絶対数値ですから，これは変更しないで下さい。

プログラムの中で，マクロ表現を用いていますが，これについては，展開した結果がリストの中に入っていますので，機械語でプログラムを移植される場合はさほど気にしなくて結構です。

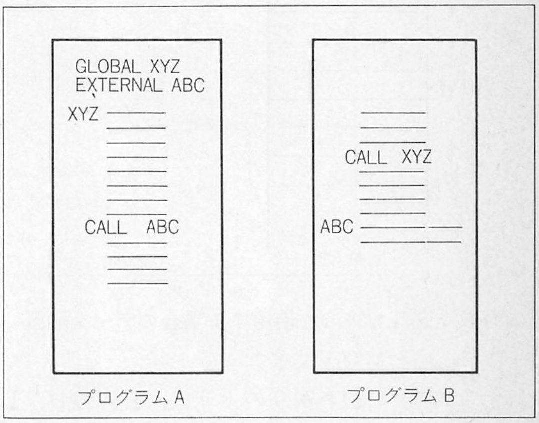
マクロを持たないアセンブラによって，プログラムの移植を行なう場合は，すこし気をつけて下さい。マクロ文は，機械語に翻訳されておりませんので，リストの左側に注意しておけばマクロ文を発見できますから，これを無視してプログラムの移植を行なって下さい。

15－4リンケージについて

各プログラムは，それぞれ小さなサブルーチンに分割されていますので，実際に演算ルーチンとして使用する場合には，これを再び結合する必要があります。

プログラムリストの先頭部分に，GLOBALおよびEXTERNALと書かれている部分がありますが，手持ちのアセンブラで，これらの表現が許されている場合には，この表現によってプログラムの結合ができます。

あるプログラム内で，GLOBAL宣言が行なわれた場合は，その宣言されたラベルが，他のプログラムからCALLされることを示しますし，またEXTERNAL表現は，逆に，そのプログラム内で他のプログラムをCALLする場合に用います。(第15－6図を参照)



《第15－6図》EXTERNAL 表現

15－5プログラム解説

ここで紹介するすべてのプログラムで，データの受け渡たしに関して，統一した約束事を定めてあります。それは第15－4図のように被演算数ならびに演算の結果が格納されるアドレスをDEレジスタで，また演算数が格納されるアドレスをHLレジスタで示します。すなわち，四則演算については，

$$〔DE〕=〔DE〕\begin{smallmatrix}\pm\\ \times\\ \div\end{smallmatrix}〔HL〕$$

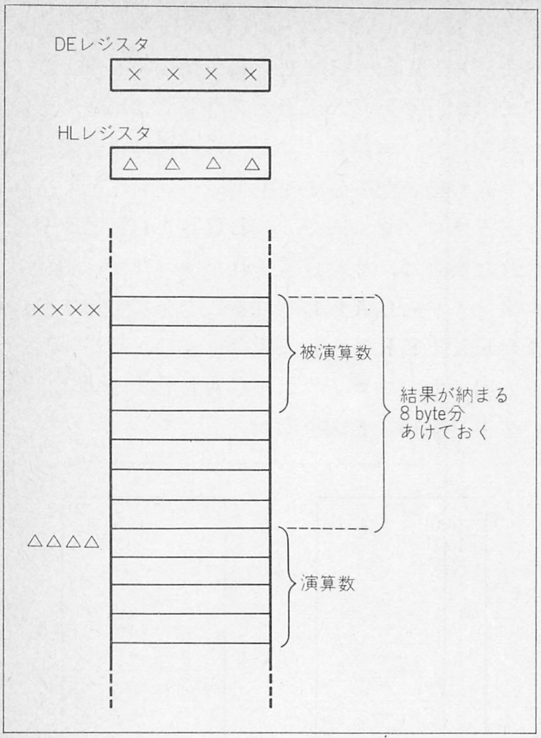
という規則に従っています。また関数演算については

$$〔DE〕=f(〔HL〕)$$

という規則に従っています。

15－5－1固定四則演算

固定四則演算を行なうプログラムは，LADD，LUB，LMLT，LDIV，LM，LD，LC



《第15－7図》LM, LDを使用する場合のデータ配置

MPおよびLMOVです。

ラベル名からおわかりのように，LADD，LSUB，LMLT，LDIVは，それぞれ多精度の加減算ならびに乗除算を行なうプログラムです。

演算の精度は，プログラムの先頭部分にあるEQU表現を変更することにより，自由に変更することができますが，一応4バイトとしてあります。例えば，データの大きさが2バイトを越えないとわかっている場合には，

```
BYTE    EQU    4  を
BYTE    EQU    2
```

と変更する事ができます。

このプログラムの使い方としては，**第15－4図**のようにあらかじめデータを特定のアドレスにセットしておき，例えば乗算の場合には，

```
LD   HL, △△△△
LD   DE, ××××
CALL LMLT
```

の様に使います。LMLTにより乗算を行なう場合に，演算結果が4バイトを越える場合がありますが，4バイトを越える部分については，丸め処理を行なった後に切り捨てています。またLDIVによる除算の結果得られた余りについては切り捨てています。

《第15－1表》フラグの変化

状　態	変化するフラグ
オーバーフロー時	Pe，M
結果が正	Z
結果が範囲外	P，NZ

これに対して，LMならびにLDは，やはりそれぞれ乗除算のサブルーチンですが，LMを使用した場合は演算の結果をすべて保存しますし，LDを使用した場合は演算の結果である商と，余りを保存しています。これらについては**第15－7図**を参照下さい。

最初に述べましたように，ここでとり扱うデータは，先頭ビットが符号を表わしているものとしています。すなわち先頭ビットが0であれば正数，1であれば負数を表わすものとしています。演算の結果はこの符号に従い，同時にフラグも変化します。これらについては**第15－1**表を参照下さい。

なお，これらのサブルーチンの中で，裏レジスタを除く全てのレジスタが破壊されます。

LCMPは，DE，およびHLレジスタでアドレス表示された二つのデータについて，その大きさを比較するルーチンで，データの大きさに従って，以下の様にフラグが変化します。尚，データそのものは破壊されずに残ります。

《第15－2表》各クラブの変化

状　態	フラグの変化
〔DE〕>〔HL〕	NZ　NC　P
〔DE〕=〔HL〕	NZ　NC　P
〔DE〕<〔HL〕	NZ　C　M

また，LMOVはデータの転送ルーチンで，その転送方向は〔HL〕→〔DE〕の方向です。このルーチンでは，HLレジスタにより示されたアドレスのデータの大きさによって，フラグが以下の様に変化します。

《第15－3表》

状　態	フラグの変化
〔HL〕>0	NZ，P
〔HL〕=0	Z，P
〔HL〕<0	NZ，M

15－5－2固定関数演算

固定関数のルーチンはＬＡＢＳ，ＬＭＯＤ，およびＬＳＧＮです。

まずＬＡＢＳは絶対値を求めるルーチンで，ＨＬレジスタにより示されたアドレスのデータの符号が正の場合にはそのまま，また負の時には正に直してからＤＥレジスタで示されるアドレスに格納します。

ＬＭＯＤは，被除数を除数で割った場合の余りを求めるルーチンで式で表わすと次のようになります。

〔DE〕=〔DE〕−(〔DE〕/〔HL〕)＊〔HL〕

また，ＬＳＧＮは，ＨＬレジスタで示されるアドレスのデータの符号に従って，ＤＥレジスタで示されるアドレスに１，０，－１を格納するルーチンです。

〔HL〕＞０　のとき　〔DE〕＝１
〔HL〕＝０　のとき　〔DE〕＝０
〔HL〕＜０　のとき　〔DE〕＝－１

15－5－3浮動四則演算

浮動四則演算を行なうプログラムはＤＡＤＤ，ＤＳＵＢ，ＤＭＬＴ，ＤＤＩＶ，ＤＣＭＰおよび，ＤＭＯＶでそれぞれ浮動小数点数の加減算，乗除算，比較ならびに転送を行なうプログラムです。

プログラムの使用方法は固定小数点数の四則演算と同一です。また，ＢＹＴＥのＥＱＵ表現を変更することによって任意の精度のデータを取り扱うことができます。

15－5－4浮動関数演算

浮動関数には，一般の三角関数や平方根，指数関数などが含まれ，演算ルーチンの中心を成す部分です。

ここでは，ＤＡＢＳ，ＤＣＯＳ，ＤＥＸＰ，ＤＦＩＸＤＬＯＧ，ＤＭＯＤ，ＤＳＧＮ，ＤＳＩＮおよびＤＳＱＲＴのプログラムを用意しております。それぞれのプログラムについて解説いたしましょう。

ＤＡＢＳ：浮動小数点数の絶対値を求めるプログラムで，処理は仮数部の符号ビットを０にしています。

ＤＦＩＸ：これはＨＬレジスタでアドレス表示された小数点以下の浮動小数点数のデータを整数化するルーチンで小数点以下の部分は切り捨てます。例えば，

$$0.1224\times16^2\rightarrow0.12\times16^2$$

のように処理を行ない，結果はＤＥレジスタで示されるアドレスに格納します。

ＤＳＩＮ：ラジアンで与えられた引数に対して，ＳＴＮを計算するルーチンで，そのアルゴリズムとして，

$$\sin x=\frac{x}{1!}-\frac{x^3}{3!}+\frac{x^5}{5!}$$

のようにテイラー級数展開を用いています。展開項数は常に固定で11項，つまりx^{23}の項まで計算を行なっています。

ＤＣＯＳ：ラジアンで与えられた引数に対して，ＣＯＳを計算するプログラムですが，このプログラム内では実際の演算は行なわず，引数に$\pi/2$を加えて，ＤＳＩＮのルーチンへこのデータを送っています。

ＤＥＸＰ：これは指数e^xを求める演算ルーチンです。

演算のアルゴリズムは，まず引数ｘに対して，ＤＦＩＸのルーチンでこれを整数化した数値x'を求め，$e^{x'}$を計算します。これはｅをx'回掛け合わせる方法で行なっています。次に$e^{x-x'}$を求めますが，これはテーラー展開

$$e^{x-x'}=1+\frac{x-x'}{1!}+\frac{(x-x')^2}{2!}+\frac{(x-x')^3}{3!}+\cdots$$

により求めています。この計算を行なう場合の展開する項数は特に定めず，展開項の大きさが1.0×10^{-17}以下になるまで計算をくり返しています。プログラムリストの中でK17と記されたラベルが，この数値を表わしています。

演算の最終結果は$e^{x'}\times e^{x-x'}$を計算し求めます。また引数が負である場合は最後に逆数を求めて，計算結果としています。

尚，このルーチンで計算できる範囲はおよそ$e^{-174}\sim e^{174}$までです。

ＤＬＯＧ：ＤＥＸＰと逆に，ｅを底とした対数を求めるルーチンです。

ＤＳＱＲＴ：平方根を求めるルーチンです。こ

のルーチンのアルゴリズムは，引数 x を次の様に変形してから計算を行ないます。

$$x = \sqrt{r \times 16^n} \quad (\text{但し} \quad 1 \leqq r < 16)$$
$$= \sqrt{r \times 4^n}$$

このように引数を分解した後，$\sqrt{r}$についてNewton-Raphson 法により計算を行ないます。

DMOD：DEレジスタでアドレス表示されるデータを，HLレジスタでアドレス表示されるデータによって割った時の余りを求めるルーチンです。以下の式に従って計算を行なっています。

$$[DE] = [DE] - [\![\underbrace{[DE]/[HL]}_{\text{整数化}}]\!] * [HL]$$

15－5－5 変換ルーチン

データの変換ルーチンとして，固定小数点数表示と浮動小数点数表示との間の変換，および，2進数データのＪＩＳコードへの変換などのプログラムを用意してあります。レジスタの使い方は他のルーチンと同じで，HLレジスタには変換しようとするデータのアドレスをDEレジスタには結果を格納しようとするアドレスをそれぞれ記入してからプログラムを呼びます。

ＬＤＦＩＸ：浮動小数点数のデータを固定小数点数（整数）に変換するルーチンです。固定小数点数のデータ領域として4バイトを指定している場合，その表わす事のできる範囲は

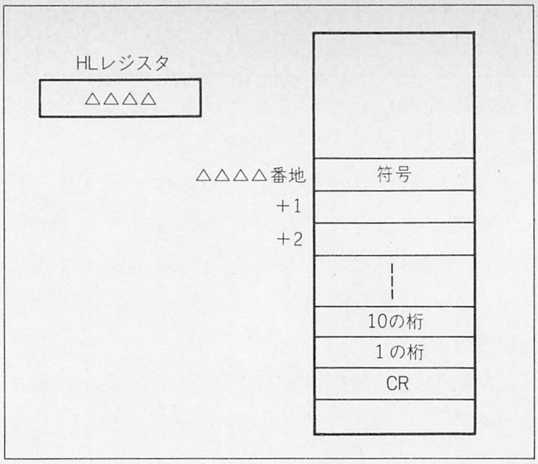
$-2147483648 \leqq X \geqq 2147483647$

の間です。これを越える数値についてはオーバーフローフラグが立ちます。また，整数に変換した際の小数点以下の部分は切り捨てています。

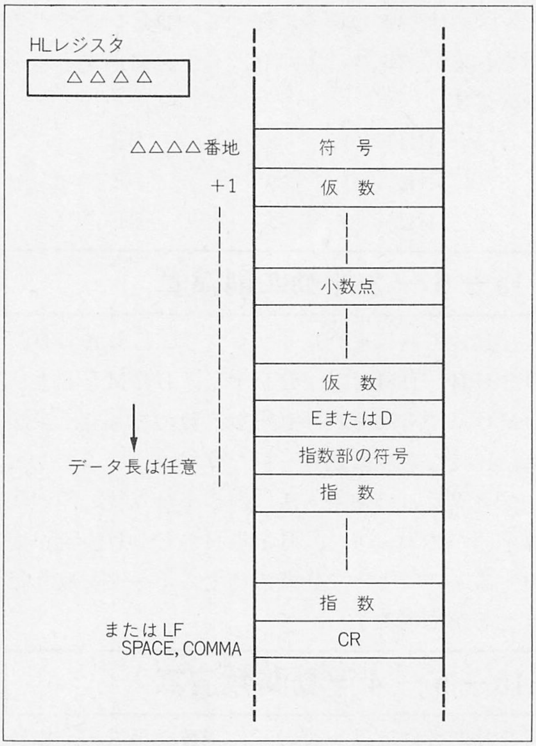
ＤＦＬＯＡＴ：ＬＤＦＩＸと逆に，固定小数点数を浮動小数点数に変換するルーチンです。

ＪＴＯＬ：ＪＩＳコードで書かれた10進数のデータを固定小数点数に変換するルーチンで，キー・ボードなどから入されたデータを2進数に変換する時に使用すると便利です。ＪＩＳコードのデータの形式は，**第15－8図**のように何バイトであってもかまいません。

データの最後のＣＲは，ＬＦ又はＳＰＡＣＥ，あるいはコンマで代用してもかまいません。変換のアルゴリズムとしては，ＪＩＳコードで書かれたデータを下位バイトから順に2進数に変換し，データを10倍しながら加え合わせています。



《第15－8図》ＪＩＳコードデータ形式



《第15－9図》ＪＴＯＤの初期データ

ＬＴＯＪ：ＪＴＯＬとは逆に，固定小数点数からＪＩＳコードへの変換ルーチンです。この変換は上位バイトから10で割って引算する事により得られます。

ＪＴＯＤ：ＪＩＳコードで書かれたデーターを浮動小数点数に変換するルーチンです。最初に与えるデータは，**第15－9図**の様な形式を持つことが必要です。仮数は0の場合には省略する事ができますが，小数点を省略するとエラーになります。

変換のアルゴリズムは，まずＪＩＳコードで書かれた仮数データを上位バイトから小数点までを順次変換し，小数点以下のデータについてはその桁数をカウントしながら，２進数に変換します。次に指数部についてＪＴＯＬと同じ方法で変換を行ない，最後にカウントした仮数部の桁数を調整してから正規化を行ないます。

ＤＴＯＪ：ＪＴＯＤと逆に，浮動小数点数をＪＩＳコードに変換するルーチンです。このルーチンでは，仮数部，指数部とも上位バイトから，その変換を行なっています。

Ｚ80は，極めて高性能なマイクロプロセッサであり，周辺チップも充実して各方面で使用されておりますが，ソフトウエア的には8080との互換性が禍いしてか，その能力が十分に引き出されてない様に思えます。特に各方面で発表されているプログラムに，Ｚ80の アセンブラーニモニックで書かれたものがまだまだ数少ない事は非常に残念に思います。

今回紹介した演算プログラムは私の所属する大学の研究室で人工衛星の軌道計算などに数多く使用されていますのでプログラムに虫はないと思いますが，あくまで試作プログラムに過ぎませんので，虫に対する責任は容謝させていただきます。

このプログラムがＺ80の，ソフトウエアの普及にいくらかでも，役に立てるならば幸いです。

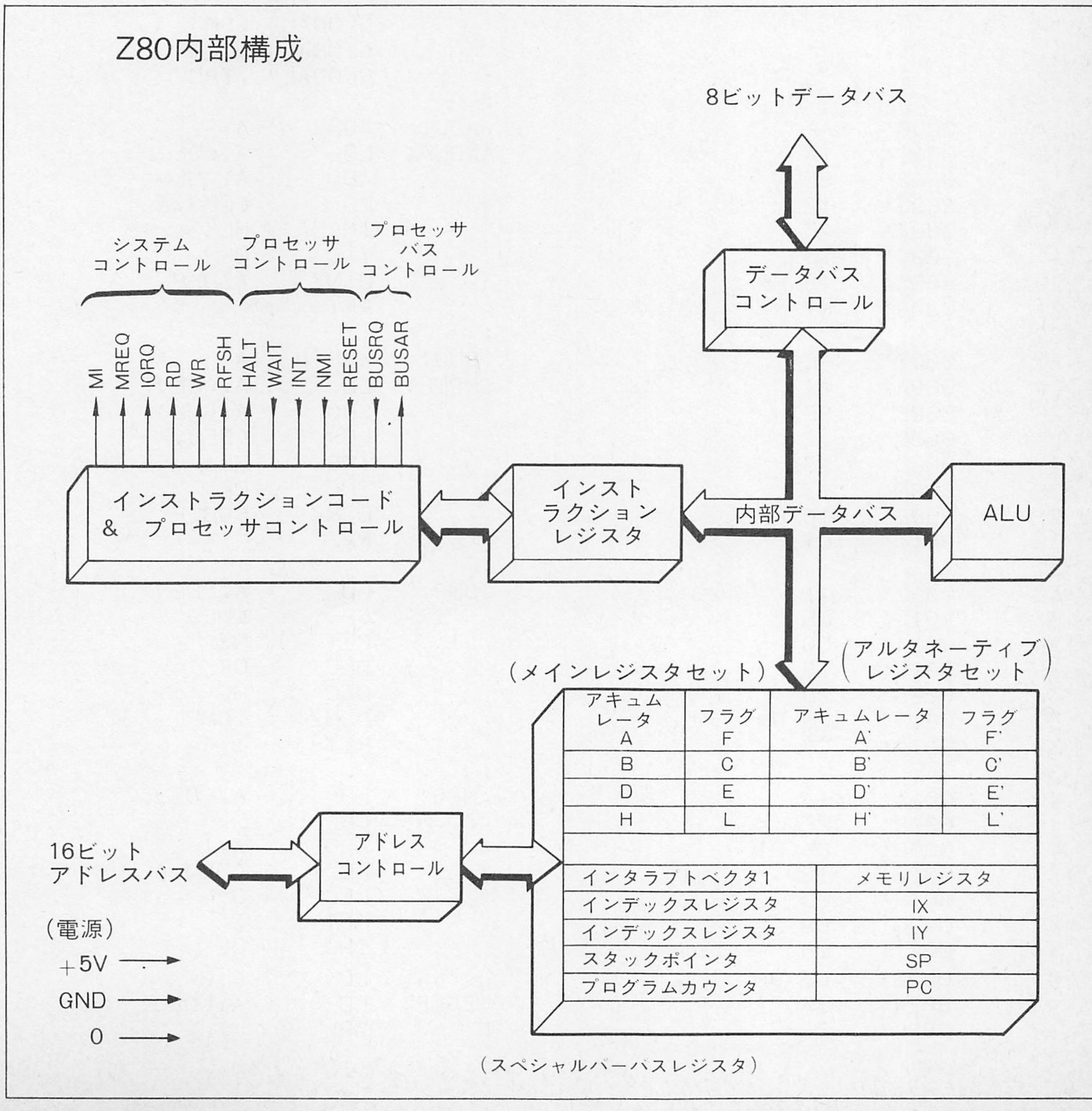
参考文献

bit　臨時増刊　マイクロコンピュータのプログラミング

共立出版

○電子計算機のための数値計算法

山内二郎　他著　　培風館



ASUB1

```
ASUB1 (V02/L01)            MACRO-80 3.22   10-Apr-79         PAGE    1

                                           TITLE    ASUB1 (V02/L01)
                                   ;
                                           GLOBAL   .ADD
                                           GLOBAL   .SUB
                                           GLOBAL   .CMP
                                           GLOBAL   .NEG
                                           GLOBAL   .SL1
                                           GLOBAL   .SL4
                                           GLOBAL   .SR1
                                           GLOBAL   .SR4
                                           GLOBAL   .CLR
                                           GLOBAL   .ZRC
                                   ;
      0000'    AF                  .ADD:   XOR      A
      0001'    1A                  ADDLP:  LD       A,(DE)
      0002'    8E                          ADC      A,(HL)
      0003'    12                          LD       (DE),A
      0004'    23                          INC      HL
      0005'    13                          INC      DE
      0006'    10 F9                       DJNZ     ADDLP
      0008'    C9                          RET
                                   ;
      0009'    AF                  .SUB:   XOR      A
      000A'    1A                  SUBLP:  LD       A,(DE)
      000B'    9E                          SBC      A,(HL)
      000C'    12                          LD       (DE),A
      000D'    23                          INC      HL
      000E'    13                          INC      DE
      000F'    10 F9                       DJNZ     SUBLP
      0011'    C9                          RET
                                   ;
      0012'    1A                  .CMP:   LD       A,(DE)
      0013'    BE                          CP       (HL)
      0014'    C0                          RET      NZ
      0015'    1B                          DEC      DE
      0016'    2B                          DEC      HL
      0017'    10 F9                       DJNZ     .CMP
      0019'    C9                          RET
                                   ;
      001A'    1A                  .NEG:   LD       A,(DE)
      001B'    2F                          CPL
      001C'    C6 01                       ADD      A,1
      001E'    12                          LD       (DE),A
      001F'    05                          DEC      B
      0020'    C8                          RET      Z
      0021'    13                          INC      DE
      0022'    0E 00                       LD       C,0
      0024'    1A                  NEGLP:  LD       A,(DE)
      0025'    2F                          CPL
```

```
        0026'   89                      ADC     A,C
        0027'   12                      LD      (DE),A
        0028'   13                      INC     DE
        0029'   10 F9                   DJNZ    NEGLP
        002B'   C9                      RET
                                ;
        002C'   AF              .SL1:   XOR     A
        002D'   CB 16           SL1LP:  RL      (HL)
        002F'   23                      INC     HL
        0030'   10 FB                   DJNZ    SL1LP
        0032'   C9                      RET
                                ;
        0033'   AF              .SR1:   XOR     A
        0034'   CB 1E           SR1LP:  RR      M
        0036'   2B                      DEC     HL
        0037'   10 FB                   DJNZ    SR1LP
        0039'   C9                      RET
                                ;
        003A'   AF              .SL4:   XOR     A
        003B'   ED 6F           SL4LP:  RLD
        003D'   23                      INC     HL
        003E'   10 FB                   DJNZ    SL4LP
        0040'   C9                      RET
                                ;
        0041'   AF              .SR4:   XOR     A
        0042'   ED 67           SR4LP:  RRD
        0044'   2B                      DEC     HL
        0045'   10 FB                   DJNZ    SR4LP
        0047'   C9                      RET
                                ;
        0048'   1A              .CLR:   LD      A,(DE)
        0049'   13                      INC     DE
        004A'   10 FC                   DJNZ    .CLR
        004C'   C9                      RET
                                ;
        004D'   7E              .ZRC:   LD      A,(HL)
        004E'   A7                      AND     A
        004F'   C0                      RET     NZ
        0050'   23                      INC     HL
        0051'   10 FA                   DJNZ    .ZRC
        0053'   C9                      RET
                                ;
                                        END
Macros:

Symbols:
.ADD    0000I'  .CLR    0048I'  .CMP    0012I'  .NEG    001AI'
.SL1    002CI'  .SL4    003AI'  .SR1    0033I'  .SR4    0041I'
.SUB    0009I'  .ZRC    004DI'  ADDLP   0001'   NEGLP   0024'
SL1LP   002D'   SL4LP   003B'   SR1LP   0034'   SR4LP   0042'
SUBLP   000A'


No  Fatal error(s)
```

ASUB2

```
ASUB2 (V02/L03) MACRO-80 3.22    10-Apr-79          PAGE     1

                                         TITLE    ASUB2 (V02/L03)
                                ;
                                         GLOBAL   .M10
                                         GLOBAL   .D10
                                         GLOBAL   .STP
                                         GLOBAL   .DEC
                                         EXTERNAL         .SL1
                                         EXTERNAL         .ADD
                                         EXTERNAL         .CLR
                                ;
        000D                    CR       EQU      0DH
        000A                    LF       EQU      0AH
        002C                    COMMA    EQU      2CH
        0020                    SPACE    EQU      20H
        0030                    NBASE    EQU      30H
                                ;
        0000'                   .M10     EQU      $
        0000'   22 00B5'                 LD       (DSTADR),HL
        0003'   ED 53 00B7'              LD       (WRKADR),DE
        0007'   48                       LD       C,B
        0008'   06 00                    LD       B,0
        000A'   ED 43 00B9'              LD       (BYTE),BC
        000E'   ED B0                    LDIR
        0010'   2A 00B7'                 LD       HL,(WRKADR)
        0013'   3A 00B9'                 LD       A,(BYTE)
        0016'   47                       LD       B,A
        0017'   CD 0000*                 CALL     .SL1
        001A'   2A 00B7'                 LD       HL,(WRKADR)
        001D'   ED 5B 00B5'              LD       DE,(DSTADR)
        0021'   ED 4B 00B9'              LD       BC,(BYTE)
        0025'   ED B0                    LDIR
        0027'   2A 00B7'                 LD       HL,(WRKADR)
        002A'   3A 00B9'                 LD       A,(BYTE)
        002D'   47                       LD       B,A
        002E'   CD 0000*                 CALL     .SL1
        0031'   2A 00B7'                 LD       HL,(WRKADR)
        0034'   ED 5B 00B5'              LD       DE,(DSTADR)
        0038'   3A 00B9'                 LD       A,(BYTE)
        003B'   47                       LD       B,A
        003C'   CD 0000*                 CALL     .ADD
        003F'   C9                       RET
                                ;
        0040'                   .D10     EQU      $
        0040'   22 00B5'                 LD       (DSTADR),HL
        0043'   ED 53 00B7'              LD       (WRKADR),DE
        0047'   48                       LD       C,B
        0048'   06 00                    LD       B,0
        004A'   ED 43 00B9'              LD       (BYTE),BC
        004E'   ED B0                    LDIR
```

```
0050'  AF                      XOR   A
0051'  12                      LD    (DE),A
0052'  ED 5B 00B5'             LD    DE,(DSTADR)
0056'  3A 00B9'                LD    A,(BYTE)
0059'  47                      LD    B,A
005A'  04                      INC   B
005B'  AF                      XOR   A
005C'  CD 0000*                CALL  .CLR
005F'  3A 00B9'                LD    A,(BYTE)
0062'  87                      ADD   A,A
0063'  87                      ADD   A,A
0064'  87                      ADD   A,A
0065'  47                      LD    B,A
0066'  04                      INC   B
0067'  C5              D101:   PUSH  BC
0068'  2A 00B5'                LD    HL,(DSTADR)
006B'  3A 00B9'                LD    A,(BYTE)
006E'  47                      LD    B,A
006F'  CD 0000*                CALL  .SL1
0072'  2A 00B7'                LD    HL,(WRKADR)
0075'  ED 4B 00B9'             LD    BC,(BYTE)
0079'  09                      ADD   HL,BC
007A'  7E                      LD    A,(HL)
007B'  D6 0A                   SUB   10
007D'  38 06                   JR    C,D102
007F'  77                      LD    (HL),A
0080'  2A 00B5'                LD    HL,(DSTADR)
0083'  CB C6                   SET   0,(HL)
0085'  2A 00B7'        D102:   LD    HL,(WRKADR)
0088'  3A 00B9'                LD    A,(BYTE)
008B'  47                      LD    B,A
008C'  04                      INC   B
008D'  CD 0000*                CALL  .SL1
0090'  C1                      POP   BC
0091'  10 D4                   DJNZ  D101
0093'  2A 00B7'                LD    HL,(WRKADR)
0096'  ED 4B 00B9'             LD    BC,(BYTE)
009A'  09                      ADD   HL,BC
009B'  7E                      LD    A,(HL)
009C'  CB 3F                   SRL   A
009E'  C9                      RET
                       ;
009F'                  .STP    EQU   $
009F'  FE 0D                   CP    CR
00A1'  C8                      RET   Z
00A2'  FE 0A                   CP    LF
00A4'  C8                      RET   Z
00A5'  FE 2C                   CP    COMMA
00A7'  C8                      RET   Z
00A8'  FE 20                   CP    SPACE
00AA'  C9                      RET
                       ;
00AB'                  .DEC    EQU   $
00AB'  FE 30                   CP    NBASE
00AD'  D8                      RET   C
00AE'  FE 3A                   CP    NBASE+10
00B0'  3F                      CCF
00B1'  D8                      RET   C
00B2'  E6 0F                   AND   0FH
00B4'  C9                      RET
                       ;
00B5'  0000            DSTADR: DEFW  0
00B7'  0000            WRKADR: DEFW  0
00B9'  0000            BYTE:   DEFW  0
                       ;
                               END
```

```
Macros:

Symbols:
.ADD     003D*   .CLR    005D*   .D10    0040I'  .DEC    00ABI'
.M10     0000I'  .SL1    008E*   .STP    009FI'  BYTE    00B9'
COMMA    002C    CR      000D    D101    0067'   D102    0085'
DSTADR   00B5'   LF      000A    NBASE   0030    SPACE   0020
WRKADR   00B7'


No  Fatal error(s)
```

DABS

```
DABS (V02/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1

                                                TITLE   DABS (V02/L01)
                                        ;
                                                GLOBAL  DABS
                                                EXTERNAL        DMOV
                                        ;
          0000'                         DABS    EQU     $
          0000'     ED 53 0013'                 LD      (DSTADR),DE
          0004'     CD 0000*                    CALL    DMOV
          0007'     21 0007                     LD      HL,7
          000A'     ED 5B 0013'                 LD      DE,(DSTADR)
          000E'     19                          ADD     HL,DE
          000F'     CB BE                       RES     7,(HL)
          0011'     AF                          XOR     A
          0012'     C9                          RET
                                        ;
          0013'     0000                DSTADR: DEFW    0
                                        ;
                                                END


Macros:

Symbols:
DABS    0000I'  DMOV    0005*   DSTADR  0013'


No  Fatal error(s)
```

DATAN

```
DATAN (V02/L01)           MACRO-80 3.22   10-Apr-79        PAGE     1

                                           TITLE     DATAN (V02/L01)
                                   ;
                                           GLOBAL    DATAN
                                           EXTERNAL            DSQRT
                                           EXTERNAL            DADD
                                           EXTERNAL            DSUB
                                           EXTERNAL            DCMP
                                           EXTERNAL            DMLT
                                           EXTERNAL            DDIV
                                           EXTERNAL            DMOV
                                   ;
                                   DADD    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DADD
                                           ENDM
                                   ;
                                   DSUB    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DSUB
                                           ENDM
                                   ;
                                   DCMP    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DCMP
                                           ENDM
                                   ;
                                   DMLT    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DMLT
                                           ENDM
                                   ;
                                   DDIV    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DDIV
                                           ENDM
                                   ;
                                   DMOV    MACRO     DST,SRC
                                           LD        HL,SRC
                                           LD        DE,DST
                                           CALL      DMOV
                                           ENDM
                                   ;
         0000'                     DATAN   EQU       $
         0000'    ED 53 0148'              LD        (DSTADR),DE
```

```
0004'   11 014A'            LD      DE,X
0007'   CD 0000*            CALL    DMOV
000A'   21 0151'            LD      HL,X+7
000D'   7E                  LD      A,(HL)
000E'   4F                  LD      C,A
000F'   E6 80               AND     80H
0011'   32 0147'            LD      (SGNX),A
0014'   79                  LD      A,C
0015'   E6 7F               AND     7FH
0017'   77                  LD      (HL),A
0018'   21 0152'            LD      HL,TT
001B'   06 08               LD      B,8
001D'   AF                  XOR     A
001E'   77          ATNCLR: LD      (HL),A
001F'   23                  INC     HL
0020'   10 FC               DJNZ    ATNCLR
                            DCMP    X,K1
0022'   21 0192'            LD      HL,K1
0025'   11 014A'            LD      DE,X
0028'   CD 0000*            CALL    DCMP
002B'   38 2D               JR      C,ATAN1
                            DMOV    TT,PAI04
002D'   21 01AA'            LD      HL,PAI04
0030'   11 0152'            LD      DE,TT
0033'   CD 0000*            CALL    DMOV
                            DMOV    AA,X
0036'   21 014A'            LD      HL,X
0039'   11 015A'            LD      DE,AA
003C'   CD 0000*            CALL    DMOV
                            DADD    AA,K1
003F'   21 0192'            LD      HL,K1
0042'   11 015A'            LD      DE,AA
0045'   CD 0000*            CALL    DADD
                            DSUB    X,K1
0048'   21 0192'            LD      HL,K1
004B'   11 014A'            LD      DE,X
004E'   CD 0000*            CALL    DSUB
                            DDIV    X,AA
0051'   21 015A'            LD      HL,AA
0054'   11 014A'            LD      DE,X
0057'   CD 0000*            CALL    DDIV
005A'               ATAN1   EQU     $
                            DMOV    T,K1
005A'   21 0192'            LD      HL,K1
005D'   11 0162'            LD      DE,T
0060'   CD 0000*            CALL    DMOV
0063'               ATAN2   EQU     $
                            DCMP    X,K01
0063'   21 018A'            LD      HL,K01
0066'   11 014A'            LD      DE,X
0069'   CD 0000*            CALL    DCMP
006C'   38 40               JR      C,ATAN3
                            DMLT    T,K2
006E'   21 019A'            LD      HL,K2
0071'   11 0162'            LD      DE,T
0074'   CD 0000*            CALL    DMLT
                            DMOV    AA,X
0077'   21 014A'            LD      HL,X
007A'   11 015A'            LD      DE,AA
007D'   CD 0000*            CALL    DMOV
                            DMLT    AA,X
0080'   21 014A'            LD      HL,X
0083'   11 015A'            LD      DE,AA
0086'   CD 0000*            CALL    DMLT
                            DADD    AA,K1
0089'   21 0192'            LD      HL,K1
```

```
008C'   11 015A'                LD      DE,AA
008F'   CD 0000*                CALL    DADD
0092'   21 015A'                LD      HL,AA
0095'   54                      LD      D,H
0096'   5D                      LD      E,L
0097'   CD 0000*                CALL    DSQRT
                                DADD    AA,K1
009A'   21 0192'                LD      HL,K1
009D'   11 015A'                LD      DE,AA
00A0'   CD 0000*                CALL    DADD
                                DDIV    X,AA
00A3'   21 015A'                LD      HL,AA
00A6'   11 014A'                LD      DE,X
00A9'   CD 0000*                CALL    DDIV
00AC'   18 B5                   JR      ATAN2
00AE'                   ATAN3   EQU     $
                                DMOV    X2,X
00AE'   21 014A'                LD      HL,X
00B1'   11 016A'                LD      DE,X2
00B4'   CD 0000*                CALL    DMOV
                                DMLT    X2,X
00B7'   21 014A'                LD      HL,X
00BA'   11 016A'                LD      DE,X2
00BD'   CD 0000*                CALL    DMLT
00C0'   21 0171'                LD      HL,X2+7
00C3'   CB FE                   SET     7,(HL)
                                DMOV    FN,K1
00C5'   21 0192'                LD      HL,K1
00C8'   11 0172'                LD      DE,FN
00CB'   CD 0000*                CALL    DMOV
                                DMOV    TM,K1
00CE'   21 0192'                LD      HL,K1
00D1'   11 017A'                LD      DE,TM
00D4'   CD 0000*                CALL    DMOV
                                DMOV    SM,K1
00D7'   21 0192'                LD      HL,K1
00DA'   11 0182'                LD      DE,SM
00DD'   CD 0000*                CALL    DMOV
00E0'                   ATAN4   EQU     $
                                DADD    FN,K2
00E0'   21 019A'                LD      HL,K2
00E3'   11 0172'                LD      DE,FN
00E6'   CD 0000*                CALL    DADD
                                DMLT    TM,X2
00E9'   21 016A'                LD      HL,X2
00EC'   11 017A'                LD      DE,TM
00EF'   CD 0000*                CALL    DMLT
                                DMOV    AA,TM
00F2'   21 017A'                LD      HL,TM
00F5'   11 015A'                LD      DE,AA
00F8'   CD 0000*                CALL    DMOV
                                DDIV    AA,FN
00FB'   21 0172'                LD      HL,FN
00FE'   11 015A'                LD      DE,AA
0101'   CD 0000*                CALL    DDIV
                                DADD    SM,AA
0104'   21 015A'                LD      HL,AA
0107'   11 0182'                LD      DE,SM
010A'   CD 0000*                CALL    DADD
                                DCMP    FN,K15
010D'   21 01A2'                LD      HL,K15
0110'   11 0172'                LD      DE,FN
0113'   CD 0000*                CALL    DCMP
0116'   38 C8                   JR      C,ATAN4
                                DMLT    T,SM
0118'   21 0182'                LD      HL,SM
```

```
0118'   11 0162'                         LD      DE,T
011E'   CD 0000*                         CALL    DMLT
                                         DMLT    T,X
0121'   21 014A'                         LD      HL,X
0124'   11 0162'                         LD      DE,T
0127'   CD 0000*                         CALL    DMLT
                                         DADD    T,TT
012A'   21 0152'                         LD      HL,TT
012D'   11 0162'                         LD      DE,T
0130'   CD 0000*                         CALL    DADD
0133'   21 0169'                         LD      HL,T+7
0136'   4E                               LD      C,(HL)
0137'   3A 0147'                         LD      A,(SGNX)
013A'   A9                               XOR     C
013B'   77                               LD      (HL),A
013C'   21 0162'                         LD      HL,T
013F'   ED 5B 0148'                      LD      DE,(DSTADR)
0143'   CD 0000*                         CALL    DMOV
0146'   C9                               RET
                               ;
0147'   00                     SGNX:     DEFB    0
0148'   0000                   DSTADR:   DEFW    0
014A'   00 00 00 00            X:        DEFB    0,0,0,0,0,0,0,0
014E'   00 00 00 00
0152'   00 00 00 00            TT:       DEFB    0,0,0,0,0,0,0,0
0156'   00 00 00 00
015A'   00 00 00 00            AA:       DEFB    0,0,0,0,0,0,0,0
015E'   00 00 00 00
0162'   00 00 00 00            T:        DEFB    0,0,0,0,0,0,0,0
0166'   00 00 00 00
016A'   00 00 00 00            X2:       DEFB    0,0,0,0,0,0,0,0
016E'   00 00 00 00
0172'   00 00 00 00            FN:       DEFB    0,0,0,0,0,0,0,0
0176'   00 00 00 00
017A'   00 00 00 00            TM:       DEFB    0,0,0,0,0,0,0,0
017E'   00 00 00 00
0182'   00 00 00 00            SM:       DEFB    0,0,0,0,0,0,0,0
0186'   00 00 00 00
018A'   9A 99 99 99            K01:      DEFB    9AH,99H,99H,99H
018E'   99 99 19 40                      DEFB    99H,99H,19H,40H
0192'   00 00 00 00            K1:       DEFB    0,0,0,0,0,0,10H,41H
0196'   00 00 10 41
019A'   00 00 00 00            K2:       DEFB    0,0,0,0,0,0,20H,41H
019E'   00 00 20 41
01A2'   00 00 00 00            K15:      DEFB    0,0,0,0,0,0,0F0H,41
01A6'   00 00 F0 41
01AA'   C2 68 21 A2            PAI04:    DEFB    0C2H,68H,21H,0A2H
01AE'   DA 0F C9 40                      DEFB    0DAH,0FH,0C9H,40H
                               ;

Macros:
DADD     DCMP     DDIV     DMLT     DMOV     DSUB

Symbols:
AA       015A'    ATAN1    005A'    ATAN2    0063'    ATAN3    00AE'
ATAN4    00E0'    ATNCLR   001E'    DADD     0131*    DATAN    0000I'
DCMP     0114*    DDIV     0102*    DMLT     0128*    DMOV     0144*
DSQRT    0098*    DSTADR   0148'    DSUB     004F*    FN       0172'
K01      018A'    K1       0192'    K15      01A2'    K2       019A'
PAI04    01AA'    SGNX     0147'    SM       0182'    T        0162'
TM       017A'    TT       0152'    X        014A'    X2       016A'

No  Fatal error(s)
```

DCOS

```
DCOS (V02/L01)            MACRO-80 3.22   10-Apr-79        PAGE    1

                                           TITLE    DCOS (V02/L01)
                                   ;
                                           GLOBAL   DCOS
                                           EXTERNAL         DSIN
                                           EXTERNAL         DADD
                                           EXTERNAL         DMOV
                                   ;
                                   DADD    MACRO    DST,SRC
                                           LD       HL,SRC
                                           LD       DE,DST
                                           CALL     DADD
                                           ENDM
                                   ;
        0000'                      DCOS    EQU      $
        0000'   22 0020'                   LD       (SRCADR),HL
        0003'   ED 53 0022'                LD       (DSTADR),DE
        0007'   11 0024'                   LD       DE,X
        000A'   CD 0000*                   CALL     DMOV
                                           DADD     X,PAI02
        000D'   21 002C'                   LD       HL,PAI02
        0010'   11 0024'                   LD       DE,X
        0013'   CD 0000*                   CALL     DADD
        0016'   21 0024'                   LD       HL,X
        0019'   ED 5B 0022'                LD       DE,(DSTADR)
        001D'   C3 0000*                   JP       DSIN
                                   ;
        0020'   0000               SRCADR: DEFW     0
        0022'   0000               DSTADR: DEFW     0
        0024'   00 00 00 00        X:      DEFB     0,0,0,0,0,0,0,0
        0028'   00 00 00 00
        002C'   1A 2D 44 54        PAI02:  DEFB     1AH,2DH,44H,54H
        0030'   FB 21 19 41                DEFB     0FBH,21H,19H,41
                                           END

Macros:
DADD

Symbols:
DADD    0014*   DCOS    0000I'  DMOV    000B*   DSIN    001E*
DSTADR  0022'   PAI02   002C'   SRCADR  0020'   X       0024'


No  Fatal error(s)
```

DEXP

```
DEXP (V02/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1

                                                TITLE   DEXP (V02/L01)
                                        ;
                                                GLOBAL  DEXP
                                                EXTERNAL        DFIX
                                                EXTERNAL        DADD
                                                EXTERNAL        DSUB
                                                EXTERNAL        DCMP
                                                EXTERNAL        DMLT
                                                EXTERNAL        DDIV
                                                EXTERNAL        DMOV
                                        ;
                                        DADD    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DADD
                                                ENDM
                                        ;
                                        DSUB    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DSUB
                                                ENDM
                                        ;
                                        DCMP    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DCMP
                                                ENDM
                                        ;
                                        DMLT    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DMLT
                                                ENDM
                                        ;
                                        DDIV    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DDIV
                                                ENDM
                                        ;
                                        DMOV    MACRO   DST,SRC
                                                LD      HL,SRC
                                                LD      DE,DST
                                                CALL    DMOV
                                                ENDM

        0000'                           DEXP    EQU     $
        0000'   ED 53 011C'                     LD      (DSTADR),DE
```

```
0004'   AF                      XOR     A
0005'   32 011A'                LD      (STATUS),A
0008'   11 012E'                LD      DE,X
000B'   CD 0000*                CALL    DMOV
000E'   21 0135'                LD      HL,X+7
0011'   7E                      LD      A,(HL)
0012'   E6 80                   AND     80H
0014'   32 011B'                LD      (SGNX),A
0017'   CB BE                   RES     7,(HL)
                                DCMP    X,LIMIT
0019'   21 0176'                LD      HL,LIMIT
001C'   11 012E'                LD      DE,X
001F'   CD 0000*                CALL    DCMP
0022'   F2 010A'                JP      P,OVF
0025'   21 012E'                LD      HL,X
0028'   11 011E'                LD      DE,TEMP
002B'   CD 0000*                CALL    DFIX
                                DMOV    K,K1
002E'   21 015E'                LD      HL,K1
0031'   11 014E'                LD      DE,K
0034'   CD 0000*                CALL    DMOV
0037'   21 0125'                LD      HL,TEMP+7
003A'   7E                      LD      A,(HL)
003B'   A7                      AND     A
003C'   CA 0058'                JP      Z,EXP1
003F'   2B                      DEC     HL
0040'   46                      LD      B,(HL)
0041'   FE 41                   CP      41H
0043'   20 08                   JR      NZ,EXP2
0045'   CB 38                   SRL     B
0047'   CB 38                   SRL     B
0049'   CB 38                   SRL     B
004B'   CB 38                   SRL     B
004D'                   EXP2    EQU     $
                                DMLT    K,EE
004D'   21 017E'                LD      HL,EE
0050'   11 014E'                LD      DE,K
0053'   CD 0000*                CALL    DMLT
0056'   10 F5                   DJNZ    EXP2
0058'                   EXP1    EQU     $
                                DMOV    EXPX,K1
0058'   21 015E'                LD      HL,K1
005B'   11 0136'                LD      DE,EXPX
005E'   CD 0000*                CALL    DMOV
                                DSUB    X,TEMP
0061'   21 011E'                LD      HL,TEMP
0064'   11 012E'                LD      DE,X
0067'   CD 0000*                CALL    DSUB
                                DADD    EXPX,X
006A'   21 012E'                LD      HL,X
006D'   11 0136'                LD      DE,EXPX
0070'   CD 0000*                CALL    DADD
                                DMOV    PM,K2
0073'   21 0166'                LD      HL,K2
0076'   11 013E'                LD      DE,PM
0079'   CD 0000*                CALL    DMOV
                                DMOV    LN,K2
007C'   21 0166'                LD      HL,K2
007F'   11 0146'                LD      DE,LN
0082'   CD 0000*                CALL    DMOV
                                DMOV    XX,X
0085'   21 012E'                LD      HL,X
0088'   11 0126'                LD      DE,XX
008B'   CD 0000*                CALL    DMOV
008E'                   EXPLP   EQU     $
                                DMLT    XX,X
```

```
008E'  21 012E'                 LD     HL,X
0091'  11 0126'                 LD     DE,XX
0094'  CD 0000*                 CALL   DMLT
                                DMOV   TEMP,XX
0097'  21 0126'                 LD     HL,XX
009A'  11 011E'                 LD     DE,TEMP
009D'  CD 0000*                 CALL   DMOV
                                DDIV   TEMP,PM
00A0'  21 013E'                 LD     HL,PM
00A3'  11 011E'                 LD     DE,TEMP
00A6'  CD 0000*                 CALL   DDIV
                                DADD   EXPX,TEMP
00A9'  21 011E'                 LD     HL,TEMP
00AC'  11 0136'                 LD     DE,EXPX
00AF'  CD 0000*                 CALL   DADD
                                DCMP   TEMP,K17
00B2'  21 016E'                 LD     HL,K17
00B5'  11 011E'                 LD     DE,TEMP
00B8'  CD 0000*                 CALL   DCMP
00BB'  38 14                    JR     C,EXP3
                                DADD   LN,K1
00BD'  21 015E'                 LD     HL,K1
00C0'  11 0146'                 LD     DE,LN
00C3'  CD 0000*                 CALL   DADD
                                DMLT   PM,LN
00C6'  21 0146'                 LD     HL,LN
00C9'  11 013E'                 LD     DE,PM
00CC'  CD 0000*                 CALL   DMLT
00CF'  18 BD                    JR     EXPLP
00D1'                  EXP3     EQU    $
                                DMLT   EXPX,K
00D1'  21 014E'                 LD     HL,K
00D4'  11 0136'                 LD     DE,EXPX
00D7'  CD 0000*                 CALL   DMLT
00DA'  3A 011B'                 LD     A,(SGNX)
00DD'  A7                       AND    A
00DE'  28 1B                    JR     Z,EXPRT
                                DMOV   TEMP,K1
00E0'  21 015E'                 LD     HL,K1
00E3'  11 011E'                 LD     DE,TEMP
00E6'  CD 0000*                 CALL   DMOV
                                DDIV   TEMP,EXPX
00E9'  21 0136'                 LD     HL,EXPX
00EC'  11 011E'                 LD     DE,TEMP
00EF'  CD 0000*                 CALL   DDIV
                                DMOV   EXPX,TEMP
00F2'  21 011E'                 LD     HL,TEMP
00F5'  11 0136'                 LD     DE,EXPX
00F8'  CD 0000*                 CALL   DMOV
00FB'                  EXPRT    EQU    $
                                DMOV   (DSTADR),EXPX
00FB'  21 0136'                 LD     HL,EXPX
00FE'  ED 5B 011C'              LD     DE,(DSTADR)
0102'  CD 0000*                 CALL   DMOV
0105'  3A 011A'                 LD     A,(STATUS)
0108'  B7                       OR     A
0109'  C9                       RET
010A'                  OVF      EQU    $
010A'  3E 81                    LD     A,81H
010C'  32 011A'                 LD     (STATUS),A
                                DMOV   EXPX,FMAX
010F'  21 0156'                 LD     HL,FMAX
0112'  11 0136'                 LD     DE,EXPX
0115'  CD 0000*                 CALL   DMOV
0118'  18 E1                    JR     EXPRT
                       ;
```

```
        011A'   00                      STATUS: DEFB    0
        011B'   00                      SGNX:   DEFB    0
        011C'   0000                    DSTADR: DEFW    0
        011E'   00 00 00 00             TEMP:   DEFB    0,0,0,0,0,0,0,0
        0122'   00 00 00 00
        0126'   00 00 00 00             XX:     DEFB    0,0,0,0,0,0,0,0
        012A'   00 00 00 00
        012E'   00 00 00 00             X:      DEFB    0,0,0,0,0,0,0,0
        0132'   00 00 00 00
        0136'   00 00 00 00             EXPX:   DEFB    0,0,0,0,0,0,0,0
        013A'   00 00 00 00
        013E'   00 00 00 00             PM:     DEFB    0,0,0,0,0,0,0,0
        0142'   00 00 00 00
        0146'   00 00 00 00             LN:     DEFB    0,0,0,0,0,0,0,0
        014A'   00 00 00 00
        014E'   00 00 00 00             K:      DEFB    0,0,0,0,0,0,0,0
        0152'   00 00 00 00
        0156'   FF FF FF FF             FMAX:   DEFB    0FFH,0FFH,0FFH,0FFH
        015A'   FF FF FF 7F                     DEFB    0FFH,0FFH,0FFH,7FH
        015E'   00 00 00 00             K1:     DEFB    0,0,0,0
        0162'   00 00 10 41                     DEFB    0,0,10H,41H
        0166'   00 00 00 00             K2:     DEFB    0,0,0,0
        016A'   00 00 20 41                     DEFB    0,0,20H,41H
        016E'   B4 A4 36 32             K17:    DEFB    0B4H,0A4H,36H,32H
        0172'   AA 77 B8 32                     DEFB    0AAH,77H,0B8H,32H
        0176'   3F 35 5E BA             LIMIT:  DEFB    3FH,35H,5EH,0BAH
        017A'   49 AC AE 42                     DEFB    49H,0ACH,0AEH,42H
        017E'   D2 AE 28 16             EE:     DEFB    0D2H,0AEH,28H,16H
        0182'   15 7E 2B 41                     DEFB    15H,7EH,2BH,41H
                                        ;
                                                END

Macros:
DADD    DCMP    DDIV    DMLT    DMOV    DSUB

Symbols:
DADD    00C4*   DCMP    00B9*   DDIV    00F0*   DEXP    0000I'
DFIX    002C*   DMLT    00D8*   DMOV    0116*   DSTADR  011C'
DSUB    0068*   EE      017E'   EXP1    0058'   EXP2    004D'
EXP3    00D1'   EXPLP   008E'   EXPRT   00FB'   EXPX    0136'
FMAX    0156'   K       014E'   K1      015E'   K17     016E'
K2      0166'   LIMIT   0176'   LN      0146'   OVF     010A'
PM      013E'   SGNX    011B'   STATUS  011A'   TEMP    011E'
X       012E'   XX      0126'

No  Fatal error(s)
```

DFIX

```
DFIX (V02/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1

                                                TITLE   DFIX (V02/L01)
                                        ;
                                                GLOBAL  DFIX
                                        ;
        0000'                           DFIX    EQU     $
        0000'   ED 53 004E'                     LD      (DSTADR),DE
        0004'   11 0050'                        LD      DE,TF
        0007'   01 0008                         LD      BC,8
        000A'   ED B0                           LDIR
        000C'   3A 0057'                        LD      A,(TF+7)
        000F'   E6 7F                           AND     7FH
        0011'   FE 4E                           CP      4EH
        0013'   F2 0035'                        JP      P,FIXRET
        0016'   FE 41                           CP      41H
        0018'   FA 0043'                        JP      M,FIXZ
        001B'   21 0050'                        LD      HL,TF
        001E'   4F                              LD      C,A
        001F'   3E 4E                           LD      A,4EH
        0021'   91                              SUB     C
        0022'   CB 3F                           SRL     A
        0024'   A7                              AND     A
        0025'   28 0A                           JR      Z,FIX1
        0027'   47                              LD      B,A
        0028'   AF                              XOR     A
        0029'                           FIXLP   EQU     $
        0029'   77                              LD      (HL),A
        002A'   23                              INC     HL
        002B'   10 FC                           DJNZ    FIXLP
        002D'   CB 41                           BIT     0,C
        002F'   28 04                           JR      Z,FIXRET
        0031'                           FIX1    EQU     $
        0031'   7E                              LD      A,(HL)
        0032'   E6 F0                           AND     0F0H
        0034'   77                              LD      (HL),A
        0035'                           FIXRET  EQU     $
        0035'   21 0050'                        LD      HL,TF
        0038'   ED 5B 004E'                     LD      DE,(DSTADR)
        003C'   01 0008                         LD      BC,8
        003F'   ED B0                           LDIR
        0041'   AF                              XOR     A
        0042'   C9                              RET
        0043'                           FIXZ    EQU     $
        0043'   2A 004E'                        LD      HL,(DSTADR)
        0046'   AF                              XOR     A
        0047'   06 08                           LD      B,8
        0049'                           FIXZLP  EQU     $
        0049'   77                              LD      (HL),A
        004A'   23                              INC     HL
        004B'   10 FC                           DJNZ    FIXZLP
```

```
        004D'   C9                           RET
                                  ;
        004E'   0000              DSTADR: DEFW    0
        0050'   00 00 00 00       TF:     DEFB    0,0,0,0,0,0,0,0
        0054'   00 00 00 00
                                  ;
                                          END

Macros:

Symbols:
DFIX    0000I'  DSTADR  004E'   FIX1    0031'   FIXLP   0029'
FIXRET  0035'   FIXZ    0043'   FIXZLP  0049'   TF      0050'


No  Fatal error(s)
```

DELOAT

```
DFLOAT (V02/L01)           MACRO-80 3.22   10-Apr-79        PAGE    1


                                               TITLE    DFLOAT (V02/L01)
                                       ;
                                               GLOBAL   DFLOAT
                                               EXTERNAL         .CLR
                                               EXTERNAL         .NEG
                                               EXTERNAL         .ZRC
                                               EXTERNAL         .SL4
                                       ;
         0004                          BYTEL   EQU      4
         0008                          BYTEF   EQU      8
                                       ;
         0000'                         DFLOAT  EQU      $
         0000'   3E 48                         LD       A,40H+BYTEL*2
         0002'   32 007A'                      LD       (EXPD),A
         0005'   ED 53 0078'                   LD       (DSTADR),DE
         0009'   22 0076'                      LD       (SRCADR),HL
         000C'   11 006F'                      LD       DE,WRKD
         000F'   06 03                         LD       B,BYTEF-1-BYTEL
         0011'   AF                            XOR      A
         0012'   CD 0000*                      CALL     .CLR
         0015'   21 0076'                      LD       HL,SRCADR
         0018'   01 0004                       LD       BC,BYTEL
         001B'   ED B0                         LDIR
         001D'   1B                            DEC      DE
         001E'   1A                            LD       A,(DE)
         001F'   E6 80                         AND      80H
         0021'   32 007B'                      LD       (SGND),A
         0024'   28 08                         JR       Z,DFLT1
         0026'   11 0072'                      LD       DE,WRKD+BYTEF-1-BYTE
         0029'   06 04                         LD       B,BYTEL
         002B'   CD 0000*                      CALL     .NEG
         002E'   21 0072'              DFLT1:  LD       HL,WRKD+BYTEF-1-BYTE
         0031'   06 04                         LD       B,BYTEL
         0033'   CD 0000*                      CALL     .ZRC
         0036'   28 2C                         JR       Z,DFLT2
         0038'   3A 0065'              DFLT4:  LD       A,(WRKD-BYTEF-2)
         003B'   E6 F0                         AND      0F0H
         003D'   20 0E                         JR       NZ,DFLT3
         003F'   21 0072'                      LD       HL,WRKD+BYTEF-1-BYTE
         0042'   06 04                         LD       B,BYTEL
         0044'   CD 0000*                      CALL     .SL4
         0047'   21 007A'                      LD       HL,EXPD
         004A'   35                            DEC      (HL)
         004B'   18 EB                         JR       DFLT4
         004D'   ED 5B 0078'           DFLT3:  LD       DE,(DSTADR)
         0051'   21 006F'                      LD       HL,WRKD
         0054'   01 0007                       LD       BC,BYTEF-1
         0057'   ED B0                         LDIR
         0059'   3A 007A'                      LD       A,(EXPD)
```

```
0050'   4F                          LD      C,A
005D'   3A 007B'                    LD      A,(SGND)
0060'   A9                          XOR     C
0061'   12                          LD      (DE),A
0062'   AF                          XOR     A
0063'   C9                          RET
0064'   ED 5B 0078'        DFLT2:   LD      DE,(DSTADR)
0068'   06 08                       LD      B,BYTEF
006A'   AF                          XOR     A
006B'   CD 0000*                    CALL    .CLR
006E'   C9                          RET
                           ;
006F'                      WRKD:    DEFS    BYTEF-1
0076'   0000               SRCADR:  DEFW    0
0078'   0000               DSTADR:  DEFW    0
007A'   00                 EXPD:    DEFB    0
007B'   00                 SGND:    DEFB    0
                           ;
                                    END

Macros:

Symbols:
.CLR    006C*   .NEG    002C*   .SL4    0045*   .ZRC    0034*
BYTEF   0008    BYTEL   0004    DFLOAT  0000I'  DFLT1   002E'
DFLT2   0064'   DFLT3   004D'   DFLT4   0038'   DSTADR  0078'
EXPD    007A'   SGND    007B'   SRCADR  0076'   WRKD    006F'


No  Fatal error(s)
```

D F L T

```
DFLT (V05/L01)          MACRO-80 3.22     Apr-79      PAGE    1

                                                TITLE    DFLT (V05/L01)
                                        ;
                                                GLOBAL   DADD
                                                GLOBAL   DSUB
                                                GLOBAL   DMLT
                                                GLOBAL   DDIV
                                                GLOBAL   DCMP
                                                GLOBAL   DMOV
                                                EXTERNAL          .ADD
                                                EXTERNAL          .SUB
                                                EXTERNAL          .CMP
                                                EXTERNAL          .NEG
                                                EXTERNAL          .SL1
                                                EXTERNAL          .SR1
                                                EXTERNAL          .SL4
                                                EXTERNAL          .SR4
                                                EXTERNAL          .CLR
                                                EXTERNAL          .ZRC
                                                EXTERNAL          .RND
                                        ;
         0008                           BYTE    EQU      8
                                        ;
         0000'   CD 026C'               DADD:   CALL     FLD
         0003'   3A 0317'                       LD       A,(EXPS)
         0006'   4F                             LD       C,A
         0007'   3A 0318'                       LD       A,(EXPD)
         000A'   91                             SUB      C
         000B'   DA 00DA'                       JP       C,DADD1
         000E'   FE 0F                          CP       BYTE*2-1
         0010'   D2 00C7'                       JP       NC,DADD2
         0013'   CD 02B9'               DADD12: CALL     EXPEQ
         0016'   3A 0316'               DSUB1:  LD       A,(SGND)
         0019'   A7                             AND      A
         001A'   28 08                          JR       Z,DADD3
         001C'   11 0301'                       LD       DE,WRKD
         001F'   06 0F                          LD       B,BYTE*2-1
         0021'   CD 0000*                       CALL     .NEG
         0024'   3A 0315'               DADD3:  LD       A,(SGNS)
         0027'   A7                             AND      A
         0028'   28 08                          JR       Z,DADD4
         002A'   11 02F1'                       LD       DE,WRKS
         002D'   06 0F                          LD       B,BYTE*2-1
         002F'   CD 0000*                       CALL     .NEG
         0032'   11 0301'               DADD4:  LD       DE,WRKD
         0035'   21 02F1'                       LD       HL,WRKS
         0038'   06 0F                          LD       B,BYTE*2-1
         003A'   CD 0000*                       CALL     .ADD
         003D'   3A 030F'                       LD       A,(WRKD+(BYTE-1)*2
         0040'   E6 80                          AND      80H
```

```
0042'   32 0316'               LD     (SGND),A
0045'   28 08                  JR     Z,DADD5
0047'   11 0301'               LD     DE,WRKD
004A'   06 0F                  LD     B,BYTE*2-1
004C'   CD 0000*               CALL   .NEG
004F'   3A 030F'      DADD5:   LD     A,(WRKD+(BYTE-1)*2)
0052'   A7                     AND    A
0053'   28 16                  JR     Z,DADD6
0055'   21 0318'               LD     HL,EXPD
0058'   34                     INC    (HL)
0059'   21 030F'               LD     HL,WRKD+(BYTE-1)*2
005C'   06 09                  LD     B,BYTE+1
005E'   CD 0000*               CALL   .SR4
0061'   11 0307'               LD     DE,WRKD+BYTE-2
0064'   06 08                  LD     B,BYTE
0066'   CD 0000*               CALL   .RND
0069'   18 E4                  JR     DADD5
006B'   21 0301'      DADD6:   LD     HL,WRKD
006E'   06 0F                  LD     B,BYTE*2-1
0070'   CD 0000*               CALL   .ZRC
0073'   28 54                  JR     Z,DADD7
0075'   3A 030E'      DADD9:   LD     A,(WRKD+(BYTE-1)*2-1)
0078'   E6 F0                  AND    0F0H
007A'   20 14                  JR     NZ,DADD8
007C'   3A 0318'               LD     A,(EXPD)
007F'   A7                     AND    A
0080'   28 47                  JR     Z,DADD7
0082'   3D                     DEC    A
0083'   32 0318'               LD     (EXPD),A
0086'   21 0301'               LD     HL,WRKD
0089'   06 0E                  LD     B,0+(BYTE-1)*2
008B'   CD 0000*               CALL   .SL4
008E'   18 E5                  JR     DADD9
0090'   11 0307'      DADD8:   LD     DE,WRKD+BYTE-2
0093'   06 08                  LD     B,BYTE
0095'   CD 0000*               CALL   .RND
0098'   3A 030F'               LD     A,(WRKD+(BYTE-1)*2)
009B'   A7                     AND    A
009C'   28 0E                  JR     Z,DADD10
009E'   21 0318'               LD     HL,EXPD
00A1'   34                     INC    (HL)
00A2'   21 030F'               LD     HL,WRKD+(BYTE-1)*2
00A5'   06 09                  LD     B,BYTE+1
00A7'   CD 0000*               CALL   .SR1
00AA'   18 E4                  JR     DADD8
00AC'   3A 0318'      DADD10:  LD     A,(EXPD)
00AF'   A7                     AND    A
00B0'   FA 00EE'               JP     M,OVF
00B3'   ED 5B 0313'   DADD11:  LD     DE,(DSTADR)
00B7'   21 0308'               LD     HL,WRKD+BYTE-1
00BA'   06 07                  LD     B,BYTE-1
00BC'   ED B0                  LDIR
00BE'   3A 0318'               LD     A,(EXPD)
00C1'   4F                     LD     C,A
00C2'   3A 0316'               LD     A,(SGND)
00C5'   A9                     XOR    C
00C6'   12                     LD     (DE),A
00C7'   AF            DADD2:   XOR    A
00C8'   C9                     RET
00C9'   AF            DADD7:   XOR    A
00CA'   32 0316'               LD     (SGND),A
00CD'   32 0318'               LD     (EXPD),A
00D0'   11 0308'               LD     DE,WRKD+BYTE-1
00D3'   06 07                  LD     B,BYTE-1
00D5'   CD 0000*               CALL   .CLR
00D8'   18 D9                  JR     DADD11
```

```
00DA'   ED 44                DD1:    NEG
00DC'   FE 0F                        CP      BYTE*2-1
00DE'   DA 0013'                     JP      C,DADD12
00E1'   2A 0311'                     LD      HL,(SRCADR)
00E4'   ED 5B 0313'                  LD      DE,(DSTADR)
00E8'   06 08                        LD      B,BYTE
00EA'   ED B0                        LDIR
00EC'   18 D9                        JR      DADD2
00EE'   3E 81                OVF:    LD      A,81H
00F0'   A7                           AND     A
00F1'   C9                           RET
                             ;
00F2'   CD 026C'             DSUB:   CALL    FLD
00F5'   3A 0317'                     LD      A,(EXPS)
00F8'   4F                           LD      C,A
00F9'   3A 0318'                     LD      A,(EXPD)
00FC'   91                           SUB     C
00FD'   38 13                        JR      C,DSUB2
00FF'   FE 0F                        CP      BYTE*2-1
0101'   D2 00C7'                     JP      NC,DADD2
0104'   CD 02B9'             DSUB3:  CALL    EXPEQ
0107'   11 02F1'                     LD      DE,WRKS
010A'   06 0F                        LD      B,BYTE*2-1
010C'   CD 0000*                     CALL    .NEG
010F'   C3 0016'                     JP      DSUB1
0112'   ED 44                DSUB2:  NEG
0114'   FE 0F                        CP      BYTE*2-1
0116'   DA 0104'                     JP      C,DSUB3
0119'   2A 0311'                     LD      HL,(SRCADR)
011C'   ED 5B 0313'                  LD      DE,(DSTADR)
0120'   01 0008                      LD      BC,BYTE
0123'   ED B0                        LDIR
0125'   15                           DEC     D
0126'   1A                           LD      A,(DE)
0127'   EE 80                        XOR     80H
0129'   12                           LD      (DE),A
012A'   C3 00C7'                     JP      DADD2
                             ;
012D'   CD 026C'             DCMP:   CALL    FLD
0130'   3A 0315'                     LD      A,(SGNS)
0133'   4F                           LD      C,A
0134'   3A 0316'                     LD      A,(SGND)
0137'   A9                           XOR     C
0138'   20 1B                        JR      NZ,DCMP1
013A'   3A 0317'                     LD      A,(EXPS)
013D'   4F                           LD      C,A
013E'   3A 0318'                     LD      A,(EXPD)
0141'   91                           SUB     C
0142'   C0                           RET     NZ
0143'   11 030E'                     LD      DE,WRKD+(BYTE-1)*2-1
0146'   21 02FE'                     LD      HL,WRKS+(BYTE-1)*2-1
0149'   06 07                        LD      B,BYTE-1
014B'   CD 0000*                     CALL    .CMP
014E'   38 29                        JR      C,DCMP4
0150'   C8                           RET     Z
0151'   3E 7F                DCMP5:  LD      A,7FH
0153'   B7                           OR      A
0154'   C9                           RET
0155'   3A 0316'             DCMP1:  LD      A,(SGND)
0158'   CB 3F                        SRL     A
015A'   47                           LD      B,A
015B'   CB 39                        SRL     C
015D'   79                           LD      A,C
015E'   90                           SUB     B
015F'   C9                           RET
0160'   3A 0318'             DCMP2:  LD      A,(EXPD)
```

```
0163'   4F                        LD      C,A
0164'   3A 0317'                  LD      A,(EXPS)
0167'   91                        SUB     C
0168'   C0                        RET     NZ
0169'   11 02FE'                  LD      DE,WRKS+(BYTE-1)*2-1
016C'   21 030E'                  LD      HL,WRKD+(BYTE-1)*2-1
016F'   06 07                     LD      B,BYTE-1
0171'   CD 0000*                  CALL    .CMP
0174'   38 03                     JR      C,DCMP4
0176'   C8                        RET     Z
0177'   18 D8                     JR      DCMP5
0179'   3E 80          DCMP4:     LD      A,80H
017B'   B7                        OR      A
017C'   37                        SCF
017D'   C9                        RET
                       ;
017E'   CD 026C'       DMLT:      CALL    FLD
0181'   3A 0315'                  LD      A,(SGNS)
0184'   2A 0316'                  LD      HL,(SGND)
0187'   AE                        XOR     (HL)
0188'   77                        LD      (HL),A
0189'   3A 0317'                  LD      A,(EXPS)
018C'   21 0318'                  LD      HL,EXPD
018F'   86                        ADD     A,(HL)
0190'   D6 40                     SUB     40H
0192'   DA 00C9'                  JP      C,DADD7
0195'   FA 00EE'                  JP      M,OVF
0198'   77                        LD      (HL),A
0199'   11 0301'                  LD      DE,WRKD
019C'   21 0308'                  LD      HL,WRKD+BYTE-1
019F'   06 07                     LD      B,BYTE-1
01A1'   AF                        XOR     A
01A2'   CD 0000*                  CALL    .CLR
01A5'   06 70                     LD      B,0+(BYTE-1)*8*2
01A7'   C5             DMLTLP:    PUSH    BC
01A8'   21 02FE'                  LD      HL,WRKS+(BYTE-1)*2-1
01AB'   06 07                     LD      B,BYTE-1
01AD'   CD 0000*                  CALL    .SR1
01B0'   30 0B                     JR      NC,DMLT1
01B2'   11 02E1'                  LD      DE,WRKT
01B5'   21 0301'                  LD      HL,WRKD
01B8'   06 0E                     LD      B,0+(BYTE-1)*2
01BA'   CD 0000*                  CALL    .ADD
01BD'   21 0301'       DMLT1:     LD      HL,WRKD
01C0'   06 0E                     LD      B,0+(BYTE-1)*2
01C2'   CD 0000*                  CALL    .SL1
01C5'   C1                        POP     BC
01C6'   10 DF                     DJNZ    DMLTLP
01C8'   11 0301'                  LD      DE,WRKD
01CB'   21 02E1'                  LD      HL,WRKT
01CE'   06 0E                     LD      B,0+(BYTE-1)*2
01D0'   ED B0                     LDIR
01D2'   C3 006B'                  JP      DADD6
                       ;
01D5'   CD 026C'       DDIV:      CALL    FLD
01D8'   3A 0315'                  LD      A,(SGNS)
01DB'   21 0316'                  LD      HL,SGND
01DE'   AE                        XOR     (HL)
01DF'   77                        LD      (HL),A
01E0'   3A 0317'                  LD      A,(EXPS)
01E3'   4F                        LD      C,A
01E4'   3A 0318'                  LD      A,(EXPD)
01E7'   91                        SUB     C
01E8'   C6 40                     ADD     A,40H
01EA'   F2 01F3'                  JP      P,DDIV0
01ED'   D2 00EE'                  JP      NC,OVF
```

```
01F0'   C3 00C9'                    JP      DADD7
01F3'   32 0318'        DDIV0:      LD      (EXPD),A
01F6'   ED 5B 030E'                 LD      DE,(WRKD+(BYTE-1)*2-1)
01FA'   2A 02FE'                    LD      HL,(WRKS+(BYTE-1)*2-1)
01FD'   06 07                       LD      B,BYTE-1
01FF'   CD 0000*                    CALL    .CMP
0202'   38 0E                       JR      C,DDIV1
0204'   28 0C                       JR      Z,DDIV1
0206'   2A 0318'                    LD      HL,(EXPD)
0209'   34                          INC     (HL)
020A'   21 030E'                    LD      HL,WRKD+(BYTE-1)*2-1
020D'   06 0E                       LD      B,BYTE*2-2
020F'   CD 0000*                    CALL    .SR4
0212'   06 71           DDIV1:      LD      B,0+(BYTE-1)*8*2+1
0214'   C5              DDIVLP:     PUSH    BC
0215'   21 02E1'                    LD      HL,WRKT
0218'   06 0F                       LD      B,BYTE*2-1
021A'   CD 0000*                    CALL    .SL1
021D'   11 030F'                    LD      DE,WRKD+(BYTE-1)*2
0220'   21 02FF'                    LD      HL,WRKS+(BYTE-1)*2
0223'   06 0F                       LD      B,BYTE*2-1
0225'   CD 0000*                    CALL    .CMP
0228'   38 10                       JR      C,DDIV2
022A'   21 02E1'                    LD      HL,WRKT
022D'   CB C6                       SET     0,(HL)
022F'   11 0301'                    LD      DE,WRKD
0232'   21 02F1'                    LD      HL,WRKS
0235'   06 0F                       LD      B,BYTE*2-1
0237'   CD 0000*                    CALL    .SUB
023A'   21 0301'        DDIV2:      LD      HL,WRKD
023D'   06 0F                       LD      B,BYTE*2-1
023F'   CD 0000*                    CALL    .SL1
0242'   C1                          POP     BC
0243'   10 CF                       DJNZ    DDIVLP
0245'   11 0301'                    LD      DE,WRKD
0248'   21 02E1'                    LD      HL,WRKT
024B'   01 000F                     LD      BC,BYTE*2-1
024E'   ED B0                       LDIR
0250'   C3 004F'                    JP      DADD5
0253'   22 0311'        DMOV:       LD      (SRCADR),HL
0256'   01 0008                     LD      BC,BYTE
0259'   ED B0                       LDIR
025B'   2B                          DEC     HL
025C'   7E                          LD      A,(HL)
025D'   B7                          OR      A
025E'   C0                          RET     NZ
025F'   2A 0311'                    LD      HL,(SRCADR)
0262'   06 07                       LD      B,BYTE-1
0264'   CD 0000*                    CALL    .ZRC
0267'   C8                          RET     Z
0268'   3E 7F                       LD      A,7FH
026A'   B7                          OR      A
026B'   C9                          RET
                        ;
026C'   22 0311'        FLD:        LD      (SRCADR),HL
026F'   ED 53 0313'                 LD      (DSTADR),DE
0273'   11 02E1'                    LD      DE,WRKT
0276'   06 0E                       LD      B,0+(BYTE-1)*2
0278'   AF                          XOR     A
0279'   CD 0000*                    CALL    .CLR
027C'   11 0301'                    LD      DE,WRKD
027F'   06 07                       LD      B,BYTE-1
0281'   CD 0000*                    CALL    .CLR
0284'   2A 0313'                    LD      HL,(DSTADR)
0287'   01 0007                     LD      BC,BYTE-1
028A'   ED B0                       LDIR
```

```
028C'   7E                          LD      A,(HL)
028D'   E6 7F                       AND     7FH
028F'   32 0318'                    LD      (EXPD),A
0292'   7E                          LD      A,(HL)
0293'   E6 80                       AND     80H
0295'   32 0316'                    LD      (SGND),A
0298'   AF                          XOR     A
0299'   12                          LD      (DE),A
029A'   11 02F1'                    LD      DE,WRKS
029D'   06 07                       LD      B,BYTE-1
029F'   CD 0000*                    CALL    .CLR
02A2'   2A 0311'                    LD      HL,(SRCADR)
02A5'   01 0007                     LD      BC,BYTE-1
02A8'   ED B0                       LDIR
02AA'   7E                          LD      A,(HL)
02AB'   E6 7F                       AND     7FH
02AD'   32 0317'                    LD      (EXPS),A
02B0'   7E                          LD      A,(HL)
02B1'   E6 80                       AND     80H
02B3'   32 0315'                    LD      (SGNS),A
02B6'   AF                          XOR     A
02B7'   12                          LD      (DE),A
02B8'   C9                          RET
                            ;
02B9'   3A 0317'            EXPEQ:  LD      A,(EXPS)
02BC'   4F                          LD      C,A
02BD'   3A 0318'                    LD      A,(EXPD)
02C0'   B9                          CP      C
02C1'   C8                          RET     Z
02C2'   38 0F                       JR      C,EXPEQ1
02C4'   0C                          INC     C
02C5'   79                          LD      A,C
02C6'   32 0317'                    LD      (EXPS),A
02C9'   21 02FE'                    LD      HL,WRKS+(BYTE-1)*2-1
02CC'   06 0E                       LD      B,0+(BYTE-1)*2
02CE'   CD 0000*                    CALL    .SR4
02D1'   18 E6                       JR      EXPEQ
02D3'   3C                  EXPEQ1: INC     A
02D4'   32 0318'                    LD      (EXPD),A
02D7'   21 030E'                    LD      HL,WRKD+(BYTE-1)*2-1
02DA'   06 0E                       LD      B,0+(BYTE-1)*2
02DC'   CD 0000*                    CALL    .SR4
02DF'   18 D8                       JR      EXPEQ
                            ;
02E1'                       WRKT:   DEFS    BYTE*2
02F1'                       WRKS:   DEFS    BYTE*2
0301'                       WRKD:   DEFS    BYTE*2
0311'   0000                SRCADR: DEFW    0
0313'   0000                DSTADR: DEFW    0
0315'   00                  SGNS:   DEFB    0
0316'   00                  SGND:   DEFB    0
0317'   00                  EXPS:   DEFB    0
0318'   00                  EXPD:   DEFB    0
                            ;
                                    END

Macros:

Symbols:
.ADD    01BB*   .CLR    02A0*   .CMP    0226*   .NEG    010D*
.RND    0096*   .SL1    0240*   .SL4    008C*   .SR1    01AE*
.SR4    02DD*   .SUB    0238*   .ZRC    0265*   BYTE    0008
DADD    0000I'  DADD1   00DA'   DADD10  00AC'   DADD11  00B3'
DADD12  0013'   DADD2   00C7'   DADD3   0024'   DADD4   0032'
DADD5   004F'   DADD6   006B'   DADD7   00C9'   DADD8   0090'
DADD9   0075'   DCMP    012DI'  DCMP1   0155'   DCMP2   0160'
```

```
DCMP4   0179'   DCMP5   0151'   DDIV    01D5I'  DDIV0   01F3'
DDIV1   0212'   DDIV2   023A'   DDIVLP  0214'   DMLT    017EI'
DMLT1   01BD'   DMLTLP  01A7'   DMOV    0253I'  DSTADR  0313'
DSUB    00F2I'  DSUB1   0016'   DSUB2   0112'   DSUB3   0104'
EXPD    0318'   EXPEQ   02B9'   EXPEQ1  02D3'   EXPS    0317'
.FLD    026C'   OVF     00EE'   SGND    0316'   SGNS    0315'
SRCADR  0311'   WRKD    0301'   WRKS    02F1'   WRKT    02E1'
```

```
No  Fatal error(s)
```

DLOG

```
DLOG (V02/L01)              MACRO-80 3.22   10-Apr-79       PAGE    1

                                                    TITLE   DLOG (V02/L01)
                                            ;
                                                    GLOBAL  DLOG
                                                    EXTERNAL        DSQRT
                                                    EXTERNAL        DADD
                                                    EXTERNAL        DSUB
                                                    EXTERNAL        DCMP
                                                    EXTERNAL        DMLT
                                                    EXTERNAL        DDIV
                                                    EXTERNAL        DMOV
                                            ;
                                            DADD    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DADD
                                                    ENDM
                                            ;
                                            DSUB    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DSUB
                                                    ENDM
                                            ;
                                            DCMP    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DCMP
                                                    ENDM
                                            ;
                                            DMLT    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DMLT
                                                    ENDM
                                            ;
                                            DDIV    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DDIV
                                                    ENDM
                                            ;
                                            DMOV    MACRO   DST,SRC
                                                    LD      HL,SRC
                                                    LD      DE,DST
                                                    CALL    DMOV
                                                    ENDM

        0000'                               DLOG    EQU     $
        0000'   ED 53 00F4'                         LD      (DSTADR),DE
```

```
0004'   11 00F6'                    LD      DE,X
0007'   CD 0000*                    CALL    DMOV
000A'   FA 00E6'                    JP      M,ERROR
                                    DMOV    Y,K1
000D'   21 0126'                    LD      HL,K1
0010'   11 00FE'                    LD      DE,Y
0013'   CD 0000*                    CALL    DMOV
0016'                       SQRLP   EQU     $
                                    DCMP    X,K080
0016'   21 012E'                    LD      HL,K080
0019'   11 00F6'                    LD      DE,X
001C'   CD 0000*                    CALL    DCMP
001F'   38 0B                       JR      C,SQR
                                    DCMP    X,K125
0021'   21 0136'                    LD      HL,K125
0024'   11 00F6'                    LD      DE,X
0027'   CD 0000*                    CALL    DCMP
002A'   38 13                       JR      C,CAL
002C'                       SQR     EQU     $
                                    DMLT    Y,K2
002C'   21 013E'                    LD      HL,K2
002F'   11 00FE'                    LD      DE,Y
0032'   CD 0000*                    CALL    DMLT
0035'   21 00F6'                    LD      HL,X
0038'   54                          LD      D,H
0039'   5D                          LD      E,L
003A'   CD 0000*                    CALL    DSQRT
003D'   18 D7                       JR      SQRLP
003F'                       CAL     EQU     $
                                    DMOV    FX,X
003F'   21 00F6'                    LD      HL,X
0042'   11 0106'                    LD      DE,FX
0045'   CD 0000*                    CALL    DMOV
                                    DADD    FX,K1
0048'   21 0126'                    LD      HL,K1
004B'   11 0106'                    LD      DE,FX
004E'   CD 0000*                    CALL    DADD
                                    DSUB    X,K1
0051'   21 0126'                    LD      HL,K1
0054'   11 00F6'                    LD      DE,X
0057'   CD 0000*                    CALL    DSUB
                                    DDIV    X,FX
005A'   21 0106'                    LD      HL,FX
005D'   11 00F6'                    LD      DE,X
0060'   CD 0000*                    CALL    DDIV
                                    DMOV    FX,X
0063'   21 00F6'                    LD      HL,X
0066'   11 0106'                    LD      DE,FX
0069'   CD 0000*                    CALL    DMOV
                                    DMLT    FX,K2
006C'   21 013E'                    LD      HL,K2
006F'   11 0106'                    LD      DE,FX
0072'   CD 0000*                    CALL    DMLT
                                    DMOV    FM,FX
0075'   21 0106'                    LD      HL,FX
0078'   11 010E'                    LD      DE,FM
007B'   CD 0000*                    CALL    DMOV
                                    DMOV    X2,X
007E'   21 00F6'                    LD      HL,X
0081'   11 0116'                    LD      DE,X2
0084'   CD 0000*                    CALL    DMOV
                                    DMLT    X2,X
0087'   21 00F6'                    LD      HL,X
008A'   11 0116'                    LD      DE,X2
008D'   CD 0000*                    CALL    DMLT
                                    DMOV    FN,K1
```

```
0090'   21 0126'                        LD      HL,K1
0093'   11 011E'                        LD      DE,FN
0096'   CD 0000*                        CALL    DMOV
0099'                           CALLP   EQU     $
                                        DMLT    FX,X2
0099'   21 0116'                        LD      HL,X2
009C'   11 0106'                        LD      DE,FX
009F'   CD 0000*                        CALL    DMLT
                                        DADD    FN,K2
00A2'   21 013E'                        LD      HL,K2
00A5'   11 011E'                        LD      DE,FN
00A8'   CD 0000*                        CALL    DADD
                                        DMOV    X,FX
00AB'   21 0106'                        LD      HL,FX
00AE'   11 00F6'                        LD      DE,X
00B1'   CD 0000*                        CALL    DMOV
                                        DDIV    X,FN
00B4'   21 011E'                        LD      HL,FN
00B7'   11 00F6'                        LD      DE,X
00BA'   CD 0000*                        CALL    DDIV
                                        DADD    FM,X
00BD'   21 00F6'                        LD      HL,X
00C0'   11 010E'                        LD      DE,FM
00C3'   CD 0000*                        CALL    DADD
                                        DCMP    FN,K23
00C6'   21 0146'                        LD      HL,K23
00C9'   11 011E'                        LD      DE,FN
00CC'   CD 0000*                        CALL    DCMP
00CF'   20 C8                           JR      NZ,CALLP
                                        DMLT    FM,Y
00D1'   21 00FE'                        LD      HL,Y
00D4'   11 010E'                        LD      DE,FM
00D7'   CD 0000*                        CALL    DMLT
                                        DMOV    (DSTADR),FM
00DA'   21 010E'                        LD      HL,FM
00DD'   ED 5B 00F4'                     LD      DE,(DSTADR)
00E1'   CD 0000*                        CALL    DMOV
00E4'   AF                              XOR     A
00E5'   C9                              RET
00E6'                           ERROR   EQU     $
                                        DMOV    (DSTADR),MZ
00E6'   21 014E'                        LD      HL,MZ
00E9'   ED 5B 00F4'                     LD      DE,(DSTADR)
00ED'   CD 0000*                        CALL    DMOV
00F0'   3E 01                           LD      A,01H
00F2'   A7                              AND     A
00F3'   C9                              RET
                                ;
00F4'   0000                    DSTADR: DEFW    0
00F6'   00 00 00 00             X:      DEFB    0,0,0,0,0,0,0,0
00FA'   00 00 00 00
00FE'   00 00 00 00             Y:      DEFB    0,0,0,0,0,0,0,0
0102'   00 00 00 00
0106'   00 00 00 00             FX:     DEFB    0,0,0,0,0,0,0,0
010A'   00 00 00 00
010E'   00 00 00 00             FM:     DEFB    0,0,0,0,0,0,0,0
0112'   00 00 00 00
0116'   00 00 00 00             X2:     DEFB    0,0,0,0,0,0,0,0
011A'   00 00 00 00
011E'   00 00 00 00             FN:     DEFB    0,0,0,0,0,0,0,0
0122'   00 00 00 00
0126'   00 00 00 00             K1:     DEFB    0,0,0,0
012A'   00 00 10 41                     DEFB    0,0,10H,41H
012E'   CD CC CC CC             K080:   DEFB    0CDH,0CCH,0CCH,0CCH
0132'   CC CC CC 40                     DEFB    0CCH,0CCH,0CCH,40H
0136'   00 00 00 00             K125:   DEFB    0,0,0,0
```

```
013A'  00 00 14 41              DEFB    0,0,14H,41H
013E'  00 00 00 00      K2:     DEFB    0,0,0,0
0142'  00 00 20 41              DEFB    0,0,20H,41H
0146'  00 00 00 00      K23:    DEFB    0,0,0,0
014A'  00 00 17 42              DEFB    0,0,17H,42H
014E'  FF FF FF FF      MZ:     DEFB    0FFH,0FFH,0FFH,0FFH
0152'  FF FF FF FF              DEFB    0FFH,0FFH,0FFH,0FFH
                                END

Macros:
DADD    DCMP    DDIV    DMLT    DMOV    DSUB

Symbols:
CAL     003F'   CALLP   0099'   DADD    00C4*   DCMP    00CD*
DDIV    00BB*   DLOG    0000I'  DMLT    00D8*   DMOV    00EE*
DSQRT   003B*   DSTADR  00F4'   DSUB    0058*   ERROR   00E6'
FM      010E'   FN      011E'   FX      0106'   K080    012E'
K1      0126'   K125    0136'   K2      013E'   K23     0146'
MZ      014E'   SQR     002C'   SQRLP   0016'   X       00F6'
X2      0116'   Y       00FE'


No  Fatal error(s).
```

DMOD

```
DMOD (V02/L01)              MACRO-80 3.22   10-Apr-79       PAGE    1

                                               TITLE    DMOD (V02/L01)
                                        ;
                                               GLOBAL   DMOD
                                               EXTERNAL         DFIX
                                               EXTERNAL         DSUB
                                               EXTERNAL         DMLT
                                               EXTERNAL         DDIV
                                        ;
                                        DSUB   MACRO    DST,SRC
                                               LD       HL,SRC
                                               LD       DE,DST
                                               CALL     DSUB
                                               ENDM
                                        ;
                                        DMLT   MACRO    DST,SRC
                                               LD       HL,SRC
                                               LD       DE,DST
                                               CALL     DMLT
                                               ENDM
                                        ;
                                        DDIV   MACRO    DST,SRC
                                               LD       HL,SRC
                                               LD       DE,DST
                                               CALL     DDIV
                                               ENDM
                                        ;
        0000'                           DMOD   EQU      $
        0000'   ED 53 003F'                    LD       (DSTADR),DE
        0004'   11 0041'                       LD       DE,TMS
        0007'   01 0008                        LD       BC,8
        000A'   ED B0                          LDIR
        000C'   2A 003F'                       LD       HL,(DSTADR)
        000F'   11 0049'                       LD       DE,TMD
        0012'   01 0008                        LD       BC,8
        0015'   ED B0                          LDIR
                                               DDIV     TMD,TMS
        0017'   21 0041'                       LD       HL,TMS
        001A'   11 0049'                       LD       DE,TMD
        001D'   CD 0000*                       CALL     DDIV
        0020'   28 01                          JR       Z,DMOD1
        0022'   E8                             RET      PE
        0023'                           DMOD1  EQU      $
        0023'   21 0049'                       LD       HL,TMD
        0026'   54                             LD       D,H
        0027'   5D                             LD       E,L
        0028'   CD 0000*                       CALL     DFIX
                                               DMLT     TMD,TMS
        002B'   21 0041'                       LD       HL,TMS
        002E'   11 0049'                       LD       DE,TMD
```

```
        0031'    CD 0000*                       CALL     DMLT
                                                DSUB     (DSTADR),TMD
        0034'    21 0049'                       LD       HL,TMD
        0037'    ED 5B 003F'                    LD       DE,(DSTADR)
        003B'    CD 0000*                       CALL     DSUB
        003E'    C9                             RET
                                      ;
        003F'    0000                 DSTADR:   DEFW     0
        0041'    00 00 00 00          TMS:      DEFB     0,0,0,0,0,0,0,0
        0045'    00 00 00 00
        0049'    00 00 00 00          TMD:      DEFB     0,0,0,0,0,0,0,0
        004D'    00 00 00 00
                                      ;
                                                END

Macros:
DDIV     DMLT     DSUB

Symbols:
DDIV     001E*    DFIX     0029*    DMLT     0032*    DMOD     0000I'
DMOD1    0023'    DSTADR   003F'    DSUB     003C*    TMD      0049'
TMS      0041'


No  Fatal error(s)
```

DSGN

```
DSGN (V02/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1

                                          TITLE   DSGN (V02/L01)
                                  ;
                                          GLOBAL  DSGN
                                          EXTERNAL        DMOV
                                  ;
        0000'                     DSGN    EQU     $
        0000'   ED 53 002C'               LD      (DSTADR),DE
        0004'   54                        LD      D,H
        0005'   5D                        LD      E,L
        0006'   CD 0000*                  CALL    DMOV
        0009'   2A 002C'                  LD      HL,(DSTADR)
        000C'   11 0007                   LD      DE,7
        000F'   19                        ADD     HL,DE
        0010'   3E 00                     LD      A,00H
        0012'   28 07                     JR      Z,SGN1
        0014'   3E 41                     LD      A,41H
        0016'   F2 001B'                  JP      P,SGN1
        0019'   3E C1                     LD      A,0C1H
        001B'                     SGN1    EQU     $
        001B'   77                        LD      (HL),A
        001C'   2B                        DEC     HL
        001D'   CB 27                     SLA     A
        001F'   CB 27                     SLA     A
        0021'   E6 10                     AND     10H
        0023'   77                        LD      (HL),A
        0024'   06 06                     LD      B,6
        0026'   AF                        XOR     A
        0027'                     SGNLP   EQU     $
        0027'   77                        LD      (HL),A
        0028'   2B                        DEC     HL
        0029'   10 FC                     DJNZ    SGNLP
        002B'   C9                        RET
                                  ;
        002C'   0000              DSTADR: DEFW    0
                                  ;
                                          END

Macros:

Symbols:
DMOV    0007*   DSGN    0000I'  DSTADR  002C'   SGN1    001B'
SGNLP   0027'


No  Fatal error(s)
```

DSIN

```
DSIN (V02/L01)          MACRO-80 3 22   10-Apr-79       PAGE    1


                                        TITLE   DSIN (V02/L01)

                                        GLOBAL  DSIN
                                        EXTERNAL        DFIX
                                        EXTERNAL        DADD
                                        EXTERNAL        DSUB
                                        EXTERNAL        DCMP
                                        EXTERNAL        DMLT
                                        EXTERNAL        DDIV
                                        EXTERNAL        DMOV
                                ;
                                DADD    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DADD
                                        ENDM
                                ;
                                DSUB    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DSUB
                                        ENDM
                                ;
                                DCMP    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DCMP
                                        ENDM
                                ;
                                DMLT    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DMLT
                                        ENDM
                                ;
                                DDIV    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DDIV
                                        ENDM
                                ;
                                DMOV    MACRO   DST,SRC
                                        LD      HL,SRC
                                        LD      DE,DST
                                        CALL    DMOV
                                        ENDM

        0000'                   DSIN    EQU     $
        0000'   ED 53 0153'             LD      (DSTADR),DE
```

```
0004'   11 015E'                    LD      DE,X
0007'   CD 0000*                    CALL    DMOV
000A'   21 0165'                    LD      HL,X+7
000D'   7E                          LD      A,(HL)
000E'   4F                          LD      C,A
000F'   E6 80                       AND     80H
0011'   32 015D'                    LD      (SGNX),A
0014'   79                          LD      A,C
0015'   E6 7F                       AND     7FH
0017'   77                          LD      (HL),A
                                    DMOV    X1,X
0018'   21 015E'                    LD      HL,X
001B'   11 0166'                    LD      DE,X1
001E'   CD 0000*                    CALL    DMOV
                                    DDIV    X1,PAI20
0021'   21 018E'                    LD      HL,PAI20
0024'   11 0166'                    LD      DE,X1
0027'   CD 0000*                    CALL    DDIV
002A'   21 0166'                    LD      HL,X1
002D'   54                          LD      D,H
002E'   5D                          LD      E,L
002F'   CD 0000*                    CALL    DFIX
                                    DMLT    X1,PAI20
0032'   21 018E'                    LD      HL,PAI20
0035'   11 0166'                    LD      DE,X1
0038'   CD 0000*                    CALL    DMLT
                                    DSUB    X,X1
003B'   21 0166'                    LD      HL,X1
003E'   11 015E'                    LD      DE,X
0041'   CD 0000*                    CALL    DSUB
                                    DCMP    X,PAI32
0044'   21 01A6'                    LD      HL,PAI32
0047'   11 015E'                    LD      DE,X
004A'   CD 0000*                    CALL    DCMP
004D'   F2 0103'                    JP      P,SIN1
                                    DCMP    X,PAI10
0050'   21 0196'                    LD      HL,PAI10
0053'   11 015E'                    LD      DE,X
0056'   CD 0000*                    CALL    DCMP
0059'   F2 0129'                    JP      P,SIN2
                                    DCMP    X,PAI02
005C'   21 019E'                    LD      HL,PAI02
005F'   11 015E'                    LD      DE,X
0062'   CD 0000*                    CALL    DCMP
0065'   F2 013D'                    JP      P,SIN3
0068'                       SINSUM  EQU     $
                                    DMOV    X1,X
0068'   21 015E'                    LD      HL,X
006B'   11 0166'                    LD      DE,X1
006E'   CD 0000*                    CALL    DMOV
                                    DMLT    X,X
0071'   21 015E'                    LD      HL,X
0074'   11 015E'                    LD      DE,X
0077'   CD 0000*                    CALL    DMLT
007A'   21 0165'                    LD      HL,X+7
007D'   CB FE                       SET     7,(HL)
007F'   21 016E'                    LD      HL,X2
0082'   06 08                       LD      B,8
0084'   AF                          XOR     A
0085'                       SINCLR  EQU     $
0085'   77                          LD      (HL),A
0086'   23                          INC     HL
0087'   10 FC                       DJNZ    SINCLR
                                    DMOV    X3,K1
0089'   21 01AE'                    LD      HL,K1
008C'   11 0176'                    LD      DE,X3
```

```
008F'   CD 0000*                  CALL    DMOV
                                  DMOV    X4,K1
0092'   21 01AE'                  LD      HL,K1
0095'   11 017E'                  LD      DE,X4
0098'   CD 0000*                  CALL    DMOV
009B'   06 0B                     LD      B,11
009D'                    SINLP    EQU     $
009D'   C5                        PUSH    BC
                                  DMOV    X5,X1
009E'   21 0166'                  LD      HL,X1
00A1'   11 0186'                  LD      DE,X5
00A4'   CD 0000*                  CALL    DMOV
                                  DDIV    X5,X3
00A7'   21 0176'                  LD      HL,X3
00AA'   11 0186'                  LD      DE,X5
00AD'   CD 0000*                  CALL    DDIV
                                  DADD    X2,X5
00B0'   21 0186'                  LD      HL,X5
00B3'   11 016E'                  LD      DE,X2
00B6'   CD 0000*                  CALL    DADD
00B9'   C1                        POP     BC
00BA'   10 02                     DJNZ    SINLP1
00BC'   18 2F                     JR      SINLP2
00BE'                    SINLP1   EQU     $
                                  DADD    X4,K1
00BE'   21 01AE'                  LD      HL,K1
00C1'   11 017E'                  LD      DE,X4
00C4'   CD 0000*                  CALL    DADD
                                  DMLT    X3,X4
00C7'   21 017E'                  LD      HL,X4
00CA'   11 0176'                  LD      DE,X3
00CD'   CD 0000*                  CALL    DMLT
                                  DADD    X4,K1
00D0'   21 01AE'                  LD      HL,K1
00D3'   11 017E'                  LD      DE,X4
00D6'   CD 0000*                  CALL    DADD
                                  DMLT    X3,X4
00D9'   21 017E'                  LD      HL,X4
00DC'   11 0176'                  LD      DE,X3
00DF'   CD 0000*                  CALL    DMLT
                                  DMLT    X1,X
00E2'   21 015E'                  LD      HL,X
00E5'   11 0166'                  LD      DE,X1
00E8'   CD 0000*                  CALL    DMLT
00EB'   18 B0                     JR      SINLP
00ED'                    SINLP2   EQU     $
00ED'   3A 015D'                  LD      A,(SGNX)
00F0'   4F                        LD      C,A
00F1'   21 0175'                  LD      HL,X2+7
00F4'   7E                        LD      A,(HL)
00F5'   A9                        XOR     C
00F6'   77                        LD      (HL),A
                                  DMOV    (DSTADR),X2
00F7'   21 016E'                  LD      HL,X2
00FA'   ED 5B 015B'               LD      DE,(DSTADR)
00FE'   CD 0000*                  CALL    DMOV
0101'   AF                        XOR     A
0102'   C9                        RET
0103'                    SIN1     EQU     $
                                  DMOV    X1,PAI20
0103'   21 018E'                  LD      HL,PAI20
0106'   11 0166'                  LD      DE,X1
0109'   CD 0000*                  CALL    DMOV
                                  DSUB    X1,X
010C'   21 015E'                  LD      HL,X
010F'   11 0166'                  LD      DE,X1
```

```
0112'   CD 0000*                    CALL    DSUB
                                    DMOV    X,X1
0115'   21 0166'                    LD      HL,X1
0118'   11 015E'                    LD      DE,X
011B'   CD 0000*                    CALL    DMOV
011E'   21 015D'                    LD      HL,SGNX
0121'   4E                          LD      C,(HL)
0122'   3E 80                       LD      A,80H
0124'   A9                          XOR     C
0125'   77                          LD      (HL),A
0126'   C3 0068'                    JP      SINSUM
0129'                       SIN2    EQU     $
                                    DSUB    X,PAI10
0129'   21 0196'                    LD      HL,PAI10
012C'   11 015E'                    LD      DE,X
012F'   CD 0000*                    CALL    DSUB
0132'   21 015D'                    LD      HL,SGNX
0135'   4E                          LD      C,(HL)
0136'   3E 80                       LD      A,80H
0138'   A9                          XOR     C
0139'   77                          LD      (HL),A
013A'   C3 0068'                    JP      SINSUM
013D'                       SIN3    EQU     $
                                    DMOV    X1,PAI10
013D'   21 0196'                    LD      HL,PAI10
0140'   11 0166'                    LD      DE,X1
0143'   CD 0000*                    CALL    DMOV
                                    DSUB    X1,X
0146'   21 015E'                    LD      HL,X
0149'   11 0166'                    LD      DE,X1
014C'   CD 0000*                    CALL    DSUB
                                    DMOV    X,X1
014F'   21 0166'                    LD      HL,X1
0152'   11 015E'                    LD      DE,X
0155'   CD 0000*                    CALL    DMOV
0158'   C3 0068'                    JP      SINSUM
                            ;
015B'   0000                DSTADR: DEFW    0
015D'   00                  SGNX:   DEFB    0
015E'   00 00 00 00         X:      DEFB    0,0,0,0,0,0,0,0
0162'   00 00 00 00
0166'   00 00 00 00         X1:     DEFB    0,0,0,0,0,0,0,0
016A'   00 00 00 00
016E'   00 00 00 00         X2:     DEFB    0,0,0,0,0,0,0,0
0172'   00 00 00 00
0176'   00 00 00 00         X3:     DEFB    0,0,0,0,0,0,0,0
017A'   00 00 00 00
017E'   00 00 00 00         X4:     DEFB    0,0,0,0,0,0,0,0
0182'   00 00 00 00
0186'   00 00 00 00         X5:     DEFB    0,0,0,0,0,0,0,0
018A'   00 00 00 00
018E'   5F B4 10 51         PAI20:  DEFB    5FH,0B4H,10H,51H
0192'   ED 87 64 41                 DEFB    0EDH,87H,64H,41H
0196'   2F 5A 88 A8         PAI10:  DEFB    2FH,5AH,88H,0A8H
019A'   F6 43 32 41                 DEFB    0F6H,43H,32H,41H
019E'   1A 2D 44 54         PAI02:  DEFB    1AH,2DH,44H,54H
01A2'   FB 21 19 41                 DEFB    0FBH,21H,19H,41H
01A6'   49 87 CC FC         PAI32:  DEFB    49H,87H,0CCH,0FCH
01AA'   F1 65 4B 41                 DEFB    0F1H,65H,4BH,41H
01AE'   00 00 00 00         K1:     DEFB    0,0,0,0
01B2'   00 00 10 41                 DEFB    0,0,10H,41H
                                    END

Macros:
DADD    DCMP    DDIV    DMLT    DMOV    DSUB
```

```
Symbols:
DADD     00D7*   DCMP     0063*   DDIV     00AE*   DFIX     0030*
DMLT     00E9*   DMOV     0156*   DSIN     0000I'  DSTADR   015B'
DSUB     014D*   K1       01AE'   PAI02    019E'   PAI10    0196'
PAI20    018E'   PAI32    01A6'   SGNX     015D'   SIN1     0103'
SIN2     0129'   SIN3     013D'   SINCLR   0085'   SINLP    009D'
SINLP1   00BE'   SINLP2   00ED'   SINSUM   0068'   X        015E'
X1       0166'   X2       016E'   X3       0176'   X4       017E'
X5       0186'


No  Fatal error(s)
```

DSQRT

```
DSQRT (V02/L01)          MACRO-80 3.22   10-Apr-79        PAGE    1


                                                 TITLE   DSQRT (V02/L01)

                                                 GLOBAL  DSQRT
                                                 EXTERNAL         DADD
                                                 EXTERNAL         DSUB
                                                 EXTERNAL         DCMP
                                                 EXTERNAL         DMLT
                                                 EXTERNAL         DDIV
                                                 EXTERNAL         DMOV
                                         ;
                                         DADD    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DADD
                                                 ENDM
                                         ;
                                         DSUB    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DSUB
                                                 ENDM
                                         ;
                                         DCMP    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DCMP
                                                 ENDM
                                         ;
                                         DMLT    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DMLT
                                                 ENDM
                                         ;
                                         DDIV    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DDIV
                                                 ENDM
                                         ;
                                         DMOV    MACRO   DST,SRC
                                                 LD      HL,SRC
                                                 LD      DE,DST
                                                 CALL    DMOV
                                                 ENDM
                                         ;
         0000'                           DSQRT   EQU     $
         0000'   AF                              XOR     A
         0001'   32 00BD'                        LD      (STATUS),A
```

```
0004'  ED 53 00BE'            LD     (DSTADR),DE
0008'  11 00C0'               LD     DE,X
000B'  CD 0000*               CALL   DMOV
000E'  21 00C7'               LD     HL,X+7
0011'  7E                     LD     A,(HL)
0012'  36 41                  LD     (HL),41H
0014'  A7                     AND    A
0015'  FA 00B1'               JP     M,DTERR
0018'  CA 0095'               JP     Z,DTZ
001B'                 DSQRT6  EQU    $
001B'  D6 41                  SUB    41H
001D'  4F                     LD     C,A
001E'  21 00D7'               LD     HL,K+7
0021'  CB 2F                  SRA    A
0023'  C6 41                  ADD    A,41H
0025'  77                     LD     (HL),A
0026'  2B                     DEC    HL
0027'  79                     LD     A,C
0028'  0E 10                  LD     C,10H
002A'  E6 01                  AND    01H
002C'  28 04                  JR     Z,DSQRT1
002E'  CB 01                  RLC    C
0030'  CB 01                  RLC    C
0032'                 DSQRT1  EQU    $
0032'  71                     LD     (HL),C
0033'  06 06                  LD     B,6
0035'  AF                     XOR    A
0036'                 DSQRT2  EQU    $
0036'  2B                     DEC    HL
0037'  77                     LD     (HL),A
0038'  10 FC                  DJNZ   DSQRT2
                              DMOV   XN,A0
003A'  21 00E8'               LD     HL,A0
003D'  11 00C8'               LD     DE,XN
0040'  CD 0000*               CALL   DMOV
0043'                 DSQRT3  EQU    $
                              DMOV   AN,XN
0043'  21 00C8'               LD     HL,XN
0046'  11 00D8'               LD     DE,AN
0049'  CD 0000*               CALL   DMOV
                              DMOV   XN,X
004C'  21 00C0'               LD     HL,X
004F'  11 00C8'               LD     DE,XN
0052'  CD 0000*               CALL   DMOV
                              DDIV   XN,AN
0055'  21 00D8'               LD     HL,AN
0058'  11 00C8'               LD     DE,XN
005B'  CD 0000*               CALL   DDIV
                              DADD   XN,AN
005E'  21 00D8'               LD     HL,AN
0061'  11 00C8'               LD     DE,XN
0064'  CD 0000*               CALL   DADD
                              DMLT   XN,HALF
0067'  21 00E0'               LD     HL,HALF
006A'  11 00C8'               LD     DE,XN
006D'  CD 0000*               CALL   DMLT
                              DCMP   XN,AN
0070'  21 00D8'               LD     HL,AN
0073'  11 00C8'               LD     DE,XN
0076'  CD 0000*               CALL   DCMP
0079'  38 C8                  JR     C,DSQRT3
                              DMLT   XN,K
007B'  21 00D0'               LD     HL,K
007E'  11 00C8'               LD     DE,XN
0081'  CD 0000*               CALL   DMLT
0084'  21 00C8'               LD     HL,XN
```

```
        0087'                           DSQRT8   EQU    $
        0087'    01 0008                         LD     BC,8
        008A'    ED 5B 00BE'                     LD     DE,(DSTADR)
        008E'    ED B0                           LDIR
        0090'    3A 00BD'                        LD     A,(STATUS)
        0093'    A7                              AND    A
        0094'    C9                              RET
        0095'                           DTZ      EQU    $
        0095'    21 00C0'                        LD     HL,X
        0098'    7E                              LD     A,(HL)
        0099'    23                              INC    HL
        009A'    06 06                           LD     B,6
        009C'                           DSQRT4   EQU    $
        009C'    B6                              OR     (HL)
        009D'    23                              INC    HL
        009E'    10 FC                           DJNZ   DSQRT4
        00A0'    28 06                           JR     Z,DSQRT5
        00A2'    3A 00C7'                        LD     A,(X+7)
        00A5'    C3 001B'                        JP     DSQRT6
        00A8'                           DSQRT5   EQU    $
        00A8'    21 00C0'                        LD     HL,X
        00AB'    AF                              XOR    A
        00AC'    32 00C7'                        LD     (X+7),A
        00AF'    18 D6                           JR     DSQRT8
        00B1'                           DTERR    EQU    $
        00B1'    4F                              LD     C,A
        00B2'    3E 01                           LD     A,01H
        00B4'    32 00BD'                        LD     (STATUS),A
        00B7'    79                              LD     A,C
        00B8'    E6 7F                           AND    7FH
        00BA'    C3 001B'                        JP     DSQRT6
                                        ;
        00BD'    00                     STATUS:  DEFB   0
        00BE'    0000                   DSTADR:  DEFW   0
        00C0'    00 00 00 00            X:       DEFB   0,0,0,0,0,0,0,0
        00C4'    00 00 00 00
        00C8'    00 00 00 00            XN:      DEFB   0,0,0,0,0,0,0,0
        00CC'    00 00 00 00
        00D0'    00 00 00 00            K:       DEFB   0,0,0,0,0,0,0,0
        00D4'    00 00 00 00
        00D8'    00 00 00 00            AN:      DEFB   0,0,0,0,0,0,0,0
        00DC'    00 00 00 00
        00E0'    00 00 00 00            HALF:    DEFB   0,0,0,0,0,0,80H,40H
        00E4'    00 00 80 40
        00E8'    00 00 00 00            A0:      DEFB   0,0,0,0,0,0,40H,41H
        00EC'    00 00 40 41

                                                 END
Macros:
DADD      DCMP      DDIV      DMLT      DMOV      DSUB

Symbols:
A0        00E8'     AN        00D8'     DADD      0065*     DCMP      0077*
DDIV      005C*     DMLT      0082*     DMOV      0053*     DSQRT     0000I'
DSQRT1    0032'     DSQRT2    0036'     DSQRT3    0043'     DSQRT4    009C'
DSQRT5    00A8'     DSQRT6    001B'     DSQRT8    0087'     DSTADR    00BE'
DSUB      0000*     DTERR     00B1'     DTZ       0095'     HALF      00E0'
K         00D0'     STATUS    00BD'     X         00C0'     XN        00C8'

 No  Fatal error(s)
```

DTOJ

```
DTOJ (V02/L05)  MACRO-80 3.22   10-Apr-79        PAGE    1

                                        TITLE   DTOJ (V02/L05)
                                ;
                                        GLOBAL  DTOJ
                                        EXTERNAL        .CLR
                                        EXTERNAL        .ZRC
                                        EXTERNAL        .D10
                                        EXTERNAL        .SL4
                                        EXTERNAL        .M10
                                        EXTERNAL        .SR4
                                ;
        0030                    NBASE   EQU     30H
        0020                    SPACE   EQU     20H
        002D                    MINUS   EQU     2DH
        002E                    DP      EQU     2EH
        0044                    CD      EQU     44H
        0008                    BYTE    EQU     8
        000F                    BYTEC   EQU     ((BYTE-1)*8-3)*3/10
                                ;
        0000'                   DTOJ    EQU     $
        0000'   22 01A7'                LD      (SRCADR),HL
        0003'   ED 53 01A9'             LD      (DSTADR),DE
        0007'   AF                      XOR     A
        0008'   32 01A6'                LD      (EXPD),A
        000B'   11 0194'                LD      DE,WRKT
        000E'   2A 01A7'                LD      HL,(SRCADR)
        0011'   01 0007                 LD      BC,BYTE-1
        0014'   ED B0                   LDIR
        0016'   7E                      LD      A,(HL)
        0017'   E6 7F                   AND     7FH
        0019'   32 01A5'                LD      (EXPS),A
        001C'   7E                      LD      A,(HL)
        001D'   E6 80                   AND     80H
        001F'   32 01A4'                LD      (SGND),A
        0022'   AF                      XOR     A
        0023'   12                      LD      (DE),A
        0024'   21 0194'                LD      HL,WRKT
        0027'   06 07                   LD      B,BYTE-1
        0029'   CD 0000*                CALL    .ZRC
        002C'   20 05                   JR      NZ,DTOJ4
        002E'   21 01A5'                LD      HL,EXPS
        0031'   36 41                   LD      (HL),41H
        0033'   3A 01A5'        DTOJ4:  LD      A,(EXPS)
        0036'   FE 41                   CP      41H
        0038'   38 26                   JR      C,DTOJ1
        003A'   28 47                   JR      Z,DTOJ2
        003C'   21 0194'        DTOJ3:  LD      HL,WRKT
        003F'   11 019C'                LD      DE,WRKD
        0042'   06 07                   LD      B,BYTE-1
        0044'   CD 0000*                CALL    .D10
```

```
0047'  21 01A6'            LD     HL,EXPD
004A'  34                  INC    (HL)
004B'  3A 019A'            LD     A,(WRKT+BYTE-2)
004E'  E6 F0               AND    0F0H
0050'  20 EA               JR     NZ,DTOJ3
0052'  21 0194'            LD     HL,WRKT
0055'  06 07               LD     B,BYTE-1
0057'  CD 0000*            CALL   .SL4
005A'  21 01A5'            LD     HL,EXPS
005D'  35                  DEC    (HL)
005E'  18 D3               JR     DTOJ4
0060'  21 0194'   DTOJ1:   LD     HL,WRKT
0063'  11 019C'            LD     DE,WRKD
0066'  06 08               LD     B,BYTE
0068'  CD 0000*            CALL   .M10
006B'  21 01A6'            LD     HL,EXPD
006E'  35                  DEC    (HL)
006F'  3A 019B'            LD     A,(WRKT+BYTE-1)
0072'  A7                  AND    A
0073'  28 EB               JR     Z,DTOJ1
0075'  21 019B'            LD     HL,WRKT+BYTE-1
0078'  06 08               LD     B,BYTE
007A'  CD 0000*            CALL    SR4
007D'  21 01A5'            LD     HL,EXPS
0080'  34                  INC    (HL)
0081'  18 B0               JR     DTOJ4
0083'  3A 019A'   DTOJ2:   LD     A,(WRKT+BYTE-2
0086'  FE A0               CP     0A0H
0088'  38 0F               JR     C,DTOJ5
008A'  21 0194'            LD     HL,WRKT
008D'  11 019C'            LD     DE,WRKD
0090'  06 07               LD     B,BYTE-1
0092'  CD 0000*            CALL   .D10
0095'  21 01A6'            LD     HL,EXPD
0098'  34                  INC    (HL)
0099'  2A 01A9'   DTOJ5:   LD     HL,(DSTADR)
009C'  3A 01A4'            LD     A,(SGND)
009F'  A7                  AND    A
00A0'  CA 010B'            JP     Z,DTOJ6
00A3'  36 2D               LD     (HL),MINUS
00A5'  23         DTOJ61:  INC    HL
00A6'  CD 0181'            CALL   DJCNV
00A9'  36 2E               LD     (HL),DP
00AB'  23                  INC    HL
00AC'  22 01A9'            LD     (DSTADR),HL
00AF'  06 0E               LD     B,BYTEC-1
00B1'  C5         DTOJL:   PUSH   BC
00B2'  21 0194'            LD     HL,WRKT
00B5'  11 019C'            LD     DE,WRKD
00B8'  06 08               LD     B,BYTE
00BA'  CD 0000*            CALL   .M10
00BD'  2A 01A9'            LD     HL,(DSTADR)
00C0'  CD 0181'            CALL   DJCNV
00C3'  22 01A9'            LD     (DSTADR),HL
00C6'  C1                  POP    BC
00C7'  10 E8               DJNZ   DTOJL
00C9'  2A 01A9'            LD     HL,(DSTADR)
00CC'  36 44               LD     (HL),CD
00CE'  23                  INC    HL
00CF'  3A 01A6'            LD     A,(EXPD)
00D2'  CB 7F               BIT    7,A
00D4'  20 3A               JR     NZ,DTOJ7
00D6'  36 20               LD     (HL),SPACE
00D8'  0E 00      DTOJ71:  LD     C,0
00DA'  06 0A               LD     B,10
00DC'  23                  INC    HL
```

```
00DD'   90                DTOJ9:   SUB    B
00DE'   38 03                      JR     C,DTOJ8
00E0'   0C                         INC    C
00E1'   18 FA                      JR     DTOJ9
00E3'   C6 0A             DTOJ8:   ADD    A,10
00E5'   47                         LD     B,A
00E6'   79                         LD     A,C
00E7'   EE 30                      XOR    NBASE
00E9'   77                         LD     (HL),A
00EA'   23                         INC    HL
00EB'   78                         LD     A,B
00EC'   EE 30                      XOR    NBASE
00EE'   77                         LD     (HL),A
00EF'   23                         INC    HL
00F0'   22 01A9'                   LD     (DSTADR),HL
00F3'   21 0194'                   LD     HL,WRKT
00F6'   11 019C'                   LD     DE,WRKD
00F9'   06 08                      LD     B,BYTE
00FB'   CD 0000*                   CALL   .M10
00FE'   3A 019A'                   LD     A,(WRKT+BYTE-2)
0101'   FE 50                      CP     50H
0103'   30 11                      JR     NC,DTOJC
0105'   AF                DTOJR:   XOR    A
0106'   ED 5B 01A9'                LD     DE,(DSTADR)
010A'   C9                         RET
010B'   36 20             DTOJ6:   LD     (HL),SPACE
010D'   C3 00A5'                   JP     DTOJ61
0110'   36 2D             DTOJ7:   LD     (HL),MINUS
0112'   ED 44                      NEG
0114'   18 C2                      JR     DTOJ71
0116'   06 0F             DTOJC:   LD     B,BYTEC
0118'   2A 01A9'                   LD     HL,(DSTADR)
011B'   2B                DTOJC1:  DEC    HL
011C'   7E                DTOJC2:  LD     A,(HL)
011D'   FE 2E                      CP     DP
011F'   28 FA                      JR     Z,DTOJC1
0121'   3C                         INC    A
0122'   77                         LD     (HL),A
0123'   FE 3A                      CP     NBASE+10
0125'   38 DE                      JR     C,DTOJR
0127'   36 30                      LD     (HL),NBASE
0129'   2B                         DEC    HL
012A'   10 F0                      DJNZ   DTOJC2
012C'   ED 5B 01A9'                LD     DE,(DSTADR)
0130'   1B                         DEC    DE
0131'   2A 01A9'                   LD     HL,(DSTADR)
0134'   2B                         DEC    HL
0135'   2B                         DEC    HL
0136'   01 000F                    LD     BC,BYTEC
0139'   ED B8                      LDDR
013B'   3E 31                      LD     A,NBASE+1
013D'   12                         LD     (DE),A
013E'   13                         INC    DE
013F'   1A                         LD     A,(DE)
0140'   4F                         LD     C,A
0141'   3E 2E                      LD     A,DP
0143'   12                         LD     (DE),A
0144'   79                         LD     A,C
0145'   13                         INC    DE
0146'   12                         LD     (DE),A
0147'   2A 01A9'                   LD     HL,(DSTADR)
014A'   23                         INC    HL
014B'   7E                         LD     A,(HL)
014C'   FE 20                      CP     SPACE
014E'   28 10                      JR     Z,DTOJC3
0150'   23                         INC    HL
```

```
0151'   23                   INC     HL
0152'   7E                   LD      A,(HL)
0153'   3D                   DEC     A
0154'   77                   LD      (HL),A
0155'   FE 30                CP      NBASE
0157'   30 18                JR      NC,DTOJR1
0159'   36 39                LD      (HL),NBASE+9
015B'   2B                   DEC     HL
015C'   35                   DEC     (HL)
015D'   C3 0105'             JP      DTOJR
0160'   23          DTOJC3:  INC     HL
0161'   23                   INC     HL
0162'   7E                   LD      A,(HL)
0163'   3C                   INC     A
0164'   77                   LD      (HL),A
0165'   FE 3A                CP      NBASE+10
0167'   DA 0105'             JP      C,DTOJR
016A'   36 30                LD      (HL),NBASE
016C'   2B                   DEC     HL
016D'   34                   INC     (HL)
016E'   C3 0105'             JP      DTOJR
0171'   C2 0105'    DTOJR1:  JP      NZ,DTOJR
0174'   2B                   DEC     HL
0175'   7E                   LD      A,(HL)
0176'   FE 30                CP      NBASE
0178'   C2 0105'             JP      NZ,DTOJR
017B'   2B                   DEC     HL
017C'   36 20                LD      (HL),SPACE
017E'   C3 0105'             JP      DTOJR
                    ;
0181'               DJCNV    EQU     $
0181'   11 019A'             LD      DE,WRKT+BYTE-2
0184'   1A                   LD      A,(DE)
0185'   0F                   RRCA
0186'   0F                   RRCA
0187'   0F                   RRCA
0188'   0F                   RRCA
0189'   E6 0F                AND     0FH
018B'   EE 30                XOR     NBASE
018D'   77                   LD      (HL),A
018E'   23                   INC     HL
018F'   1A                   LD      A,(DE)
0190'   E6 0F                AND     0FH
0192'   12                   LD      (DE),A
0193'   C9                   RET
                    ;
0194'               WRKT:    DEFS    BYTE
019C'               WRKD:    DEFS    BYTE
01A4'   00          SGND:    DEFB    0
01A5'   00          EXPS:    DEFB    0
01A6'   00          EXPD:    DEFB    0
01A7'   0000        SRCADR:  DEFW    0
01A9'   0000        DSTADR:  DEFW    0
                    ;
                             END
```

```
Macros:

Symbols:
.CLR     0000*    D10      0093*    M10      00FC*    SL4      0058*
.SR4     007B*    .ZRC     002A*    BYTE     0008     BYTEC    000F
CD       0044     DJCNV    0181'    DP       002E     DSTADR   01A9'
DTOJ     0000I'   DTOJ1    0060'    DTOJ2    0083'    DTOJ3    003C'
DTOJ4    0033'    DTOJ5    0099'    DTOJ6    010B'    DTOJ61   00A5'
DTOJ7    0110'    DTOJ71   00D8'    DTOJ8    00E3'    DTOJ9    00DD'
DTOJC    0116'    DTOJC1   011B'    DTOJC2   011C'    DTOJC3   0160'
```

.DTOJL	00B1'	DTOJR	0105'	DTOJR1	0171'	EXPD	01A6'
EXPS	01A5'	MINUS	002D	NBASE	0030	SGND	01A4'
SPACE	0020	SRCADR	01A7'	WRKD	019C'	WRKT	0194'

No Fatal error(s)

JTOD

```
JTOD (V02/L05)  MACRO-80 3.22   10-Apr-79        PAGE     1

                                         TITLE    JTOD (V02/L05)

                                         GLOBAL   JTOD
                                         EXTERNAL          .CLR
                                         EXTERNAL          .DEC
                                         EXTERNAL          .M10
                                         EXTERNAL          .ADD
                                         EXTERNAL          .STP
                                         EXTERNAL          .ZRC
                                         EXTERNAL          .SL4
                                         EXTERNAL          .SR4
                                         EXTERNAL          .RND
                                         EXTERNAL          .D10
                                ;
        0020                    SPACE    EQU      20H
        002D                    MINUS    EQU      2DH
        002B                    PLUS     EQU      2BH
        002E                    DP       EQU      2EH
        0044                    CD       EQU      44H
        0045                    CE       EQU      45H
        0008                    BYTE     EQU      8
        0010                    BYTEC    EQU      ((BYTE-1)*8-3)*3/10+1
                                ;
        0000'                   JTOD     EQU      $
        0000'  22 0259'                  LD       (SRCADR),HL
        0003'  ED 53 025B'               LD       (DSTADR),DE
        0007'  AF                        XOR      A
        0008'  32 0258'                  LD       (SGND),A
        000B'  32 0257'                  LD       (SGNS),A
        000E'  32 0255'                  LD       (EXPS),A
        0011'  11 024C'                  LD       DE,WRKT
        0014'  06 09                     LD       B,BYTE+1
        0016'  CD 0000*                  CALL     .CLR
        0019'  11 023A'                  LD       DE,WRKS
        001C'  06 08                     LD       B,BYTE
        001E'  CD 0000*                  CALL     .CLR
        0021'  21 0256'                  LD       HL,EXPD
        0024'  36 4E                     LD       (HL),4EH
        0026'  CD 0231'         JTOD1:   CALL     GETC
        0029'  FE 20                     CP       SPACE
        002B'  28 F9                     JR       Z,JTOD1
        002D'  FE 2B                     CP       PLUS
        002F'  28 3C                     JR       Z,JTOD2
        0031'  FE 2D                     CP       MINUS
        0033'  28 57                     JR       Z,JTOD3
        0035'  FE 2E                     CP       DP
        0037'  28 5A                     JR       Z,JTOD4
        0039'  CD 0000*                  CALL     .DEC
        003C'  DA 021B'                  JP       C,DERR
```

```
003F'   32 023A'      JTOD8:   LD     (WRKS) A
0042'   4F                     LD     C,A
0043'   21 0257'               LD     HL,SGNS
0046'   7E                     LD     A,(HL)
0047'   A7                     AND    A
0048'   20 04                  JR     NZ,JTOD5
004A'   79                     LD     A,C
004B'   A7                     AND    A
004C'   28 1F                  JR     Z,JTOD2
004E'   34            JTOD5:   INC    (HL)
004F'   21 024D'               LD     HL,WRKT+1
0052'   11 0243'               LD     DE,WRKD
0055'   06 07                  LD     B,BYTE-1
0057'   CD 0000*               CALL   .M10
005A'   21 023A'               LD     HL,WRKS
005D'   11 024D'               LD     DE,WRKT+1
0060'   06 08                  LD     B,BYTE
0062'   CD 0000*               CALL   .ADD
0065'   3A 0257'               LD     A,(SGNS)
0068'   FE 10                  CP     BYTEC
006A'   D2 00D9'               JP     NC,JTOD7
006D'   CD 0231'      JTOD2:   CALL   GETC
0070'   CD 0000*               CALL   .DEC
0073'   30 CA                  JR     NC,JTOD8
0075'   FE 2E         JTOD72:  CP     DP
0077'   28 1A                  JR     Z,JTOD4
0079'   FE 44         JTOD45:  CP     CD
007B'   CA 00E8'               JP     Z,JTOD9
007E'   FE 45                  CP     CE
0080'   CA 00E8'               JP     Z,JTOD9
0083'   CD 0000*               CALL   .STP
0086'   CA 0154'               JP     Z,JTOD10
0089'   C3 021B'               JP     DERR
008C'   21 0258'      JTOD3:   LD     HL,SGND
008F'   36 80                  LD     (HL),80H
0091'   18 DA                  JR     JTOD2
0093'   3A 0257'      JTOD4:   LD     A,(SGNS)
0096'   FE 10                  CP     BYTEC
0098'   30 34                  JR     NC,JTOD6
009A'   CD 0231'               CALL   GETC
009D'   CD 0000*               CALL   .DEC
00A0'   38 D7                  JR     C,JTOD45
00A2'   32 023A'               LD     (WRKS),A
00A5'   4F                     LD     C,A
00A6'   21 0257'               LD     HL,SGNS
00A9'   7E                     LD     A,(HL)
00AA'   A7                     AND    A
00AB'   20 04                  JR     NZ,JTOD42
00AD'   79                     LD     A,C
00AE'   A7                     AND    A
00AF'   28 17                  JR     Z,JTOD43
00B1'   34            JTOD42:  INC    (HL)
00B2'   21 024D'               LD     HL,WRKT+1
00B5'   11 0243'               LD     DE,WRKD
00B8'   06 07                  LD     B,BYTE-1
00BA'   CD 0000*               CALL   .M10
00BD'   21 023A'               LD     HL,WRKS
00C0'   11 024D'               LD     DE,WRKT+1
00C3'   06 08                  LD     B,BYTE
00C5'   CD 0000*               CALL   .ADD
00C8'   21 0255'      JTOD43:  LD     HL,EXPS
00CB'   35                     DEC    (HL)
00CC'   18 C5                  JR     JTOD4
00CE'   CD 0231'      JTOD6:   CALL   GETC
00D1'   CD 0000*               CALL   .DEC
00D4'   30 F8                  JR     NC,JTOD6
```

```
00D6'  C3 0079'            JP     JTOD45
00D9'  CD 0231'  JTOD7:    CALL   GETC
00DC'  CD 0000*            CALL   .DEC
00DF'  DA 0075'            JP     C,JTOD72
00E2'  21 0255'            LD     HL,EXPS
00E5'  34                  INC    (HL)
00E6'  18 F1               JR     JTOD7
00E8'  AF        JTOD9:    XOR    A
00E9'  32 023A'            LD     (WRKS),A
00EC'  32 0242'            LD     (WRKS+BYTE),A
00EF'  32 0257'            LD     (SGNS),A
00F2'  CD 0231'            CALL   GETC
00F5'  FE 2B               CP     PLUS
00F7'  28 2C               JR     Z,JTOD91
00F9'  FE 2D               CP     MINUS
00FB'  CA 01E6'            JP     Z,JTOD92
00FE'  CD 0000*            CALL   .DEC
0101'  DA 021B'            JP     C,DERR
0104'  4F        JTOD93:   LD     C,A
0105'  21 0257'            LD     HL,SGNS
0108'  7E                  LD     A,(HL)
0109'  A7                  AND    A
010A'  20 04               JR     NZ,JTOD94
010C'  79                  LD     A,C
010D'  A7                  AND    A
010E'  28 15               JR     Z,JTOD91
0110'  34        JTOD94:   INC    (HL)
0111'  21 023A'            LD     HL,WRKS
0114'  CB 26               SLA    (HL)
0116'  7E                  LD     A,(HL)
0117'  CB 26               SLA    (HL)
0119'  CB 26               SLA    (HL)
011B'  86                  ADD    A,(HL)
011C'  81                  ADD    A,C
011D'  77                  LD     (HL),A
011E'  3A 0257'            LD     A,(SGNS)
0121'  FE 03               CP     3
0123'  30 10               JR     NC,JTOD95
0125'  CD 0231'  JTOD91:   CALL   GETC
0128'  CD 0000*            CALL   .DEC
012B'  30 D7               JR     NC,JTOD93
012D'  CD 0000*            CALL   .STP
0130'  28 0C               JR     Z,JTOD11
0132'  C3 021B'            JP     DERR
0135'  CD 0231'  JTOD95:   CALL   GETC
0138'  CD 0000*            CALL   .STP
013B'  C2 021B'            JP     NZ,DERR
013E'  21 023A'  JTOD11:   LD     HL,WRKS
0141'  3A 0242'            LD     A,(WRKS+BYTE)
0144'  A7                  AND    A
0145'  28 04               JR     Z,JTOD12
0147'  7E                  LD     A,(HL)
0148'  ED 44               NEG
014A'  77                  LD     (HL),A
014B'  11 0255'  JTOD12:   LD     DE,EXPS
014E'  1A                  LD     A,(DE)
014F'  86                  ADD    A,(HL)
0150'  12                  LD     (DE),A
0151'  EA 0220'            JP     PE,DOVF
0154'  21 024D'  JTOD10:   LD     HL,WRKT+1
0157'  06 07               LD     B,BYTE-1
0159'  CD 0000*            CALL   .ZRC
015C'  CA 0225'            JP     Z,DUNDF
015F'  3A 0253'  JTOD13:   LD     A,(WRKT+BYTE-1)
0162'  E6 F0               AND    0F0H
0164'  20 0E               JR     NZ,JTOD14
```

```
0166'   21 024D'                LD      HL,WRKT+1
0169'   06 07                   LD      B,BYTE-1
016B'   CD 0000*                CALL    .SL4
016E'   21 0256'                LD      HL,EXPD
0171'   35                      DEC     (HL)
0172'   18 EB                   JR      JTOD13
0174'   AF              JTOD14: XOR     A
0175'   32 0254'                LD      (WRKT+BYTE),A
0178'   3A 0255'                LD      A,(EXPS)
017B'   A7                      AND     A
017C'   FA 01EE'                JP      M,JTOD15
017F'   28 29                   JR      Z,JTOD16
0181'   21 024D'        JTOD17: LD      HL,WRKT+1
0184'   11 0243'                LD      DE,WRKD
0187'   06 07                   LD      B,BYTE-1
0189'   CD 0000*                CALL    .M10
018C'   3A 0254'                LD      A,(WRKT+BYTE)
018F'   A7                      AND     A
0190'   28 12                   JR      Z,JTOD18
0192'   21 0254'                LD      HL,WRKT+BYTE
0195'   06 09                   LD      B,BYTE+1
0197'   CD 0000*                CALL    .SR4
019A'   21 0256'                LD      HL,EXPD
019D'   7E                      LD      A,(HL)
019E'   C6 01                   ADD     A,1
01A0'   FA 0220'                JP      M,DOVF
01A3'   77                      LD      (HL),A
01A4'   21 0255'        JTOD18: LD      HL,EXPS
01A7'   35                      DEC     (HL)
01A8'   20 D7                   JR      NZ,JTOD17
01AA'   11 024C'        JTOD16: LD      DE,WRKT
01AD'   06 09                   LD      B,BYTE+1
01AF'   CD 0000*                CALL    .RND
01B2'   3A 0254'                LD      A,(WRKT+BYTE)
01B5'   A7                      AND     A
01B6'   28 14                   JR      Z,JTOD19
01B8'   21 0254'                LD      HL,WRKT+BYTE
01BB'   06 09                   LD      B,BYTE+1
01BD'   CD 0000*                CALL    .SR4
01C0'   21 0256'                LD      HL,EXPD
01C3'   7E                      LD      A,(HL)
01C4'   C6 01                   ADD     A,1
01C6'   FA 0220'                JP      M,DOVF
01C9'   77                      LD      (HL),A
01CA'   18 DE                   JR      JTOD16
01CC'   21 024D'        JTOD19: LD      HL,WRKT+1
01CF'   ED 5B 025B'             LD      DE,(DSTADR)
01D3'   01 0007                 LD      BC,BYTE-1
01D6'   ED B0                   LDIR
01D8'   3A 0256'                LD      A,(EXPD)
01DB'   4F                      LD      C,A
01DC'   3A 0258'                LD      A,(SGND)
01DF'   A9                      XOR     C
01E0'   12                      LD      (DE),A
01E1'   AF                      XOR     A
01E2'   2A 0259'        JTODR:  LD      HL,(SRCADR)
01E5'   C9                      RET
01E6'   21 0242'        JTOD92: LD      HL,WRKS+BYTE
01E9'   36 80                   LD      (HL),80H
01EB'   C3 0125'                JP      JTOD91
01EE'   21 024C'        JTOD15: LD      HL,WRKT
01F1'   11 0243'                LD      DE,WRKD
01F4'   06 08                   LD      B,BYTE
01F6'   CD 0000*                CALL    .D10
01F9'   3A 0253'                LD      A,(WRKT+BYTE-1)
01FC'   E6 F0                   AND     0F0H
```

```
01FE'   20 12                      JR      NZ,JTOD20
0200'   21 024C'                   LD      HL,WRKT
0203'   06 08                      LD      B,BYTE
0205'   CD 0000*                   CALL    .SL4
0208'   21 0256'                   LD      HL,EXPD
020B'   7E                         LD      A,(HL)
020C'   D6 01                      SUB     1
020E'   FA 0225'                   JP      M,DUNDF
0211'   77                         LD      (HL),A
0212'   21 0255'         JTOD20:   LD      HL,EXPS
0215'   34                         INC     (HL)
0216'   20 D6                      JR      NZ,JTOD15
0218'   C3 01AA'                   JP      JTOD16
021B'   3E 01            DERR:     LD      A,1
021D'   A7                         AND     A
021E'   18 C2                      JR      JTODR
0220'   3E 81            DOVF:     LD      A,81H
0222'   A7                         AND     A
0223'   18 BD                      JR      JTODR
0225'   ED 5B 025B'      DUNDF:    LD      DE,(DSTADR)
0229'   06 08                      LD      B,BYTE
022B'   AF                         XOR     A
022C'   CD 0000*                   CALL    .CLR
022F'   18 B1                      JR      JTODR

                         ;
0231'                    GETC      EQU     $
0231'   2A 0259'                   LD      HL,(SRCADR)
0234'   7E                         LD      A,(HL)
0235'   23                         INC     HL
0236'   22 0259'                   LD      (SRCADR),HL
0239'   C9                         RET

                         ;
023A'                    WRKS:     DEFS    BYTE+1
0243'                    WRKD:     DEFS    BYTE+1
024C'                    WRKT:     DEFS    BYTE+1
0255'   00               EXPS:     DEFB    0
0256'   00               EXPD:     DEFB    0
0257'   00               SGNS:     DEFB    0
0258'   00               SGND:     DEFB    0
0259'   0000             SRCADR:   DEFW    0
025B'   0000             DSTADR:   DEFW    0
                         ;
                                   END

Macros:

Symbols:
.ADD     00C6*    .CLR     022D*    .D10     01F7*    .DEC     0129*
.M10     018A*    .RND     01B0*    .SL4     0206*    .SR4     01BE*
.STP     0139*    .ZRC     015A*    BYTE     0008     BYTEC    0010
CD       0044     CE       0045     DERR     021B'    DOVF     0220'
DP       002E     DSTADR   025B'    DUNDF    0225'    EXPD     0256'
EXPS     0255'    GETC     0231'    JTOD     0000I'   JTOD1    0026'
JTOD10   0154'    JTOD11   013E'    JTOD12   014B'    JTOD13   015F'
JTOD14   0174'    JTOD15   01EE'    JTOD16   01AA'    JTOD17   0181'
JTOD18   01A4'    JTOD19   01CC'    JTOD2    006D'    JTOD20   0212'
JTOD3    008C'    JTOD4    0093'    JTOD42   00B1'    JTOD43   00C8'
JTOD45   0079'    JTOD5    004E'    JTOD6    00CE'    JTOD7    00D9'
JTOD72   0075'    JTOD8    003F'    JTOD9    00E8'    JTOD91   0125'
JTOD92   01E6'    JTOD93   0104'    JTOD94   0110'    JTOD95   0135'
JTODR    01E2'    MINUS    002D     PLUS     002B     SGND     0258'
SGNS     0257'    SPACE    0020     SRCADR   0259'    WRKD     0243'
WRKS     023A'    WRKT     024C'


No  Fatal error(s)
```

JTOL

```
JTOL (V02/L01)  MACRO-80 3.22   10-Apr-79        PAGE    1

                                  TITLE   JTOL (V02/L01)
                         ;
                                  GLOBAL  JTOL
                                  EXTERNAL        .CLR
                                  EXTERNAL        .DEC
                                  EXTERNAL        .STP
                                  EXTERNAL        .M10
                                  EXTERNAL        .ADD
                                  EXTERNAL        .NEG
                         ;
0020                     SPACE    EQU     20H
002B                     PLUS     EQU     2BH
002D                     MINUS    EQU     2DH
0004                     BYTE     EQU     4
                         ;
0000'                    JTOL     EQU     $
0000'    22 00AE'                 LD      (SRCADR),HL
0003'    ED 53 00B0'              LD      (DSTADR),DE
0007'    11 009E'                 LD      DE,WRKT
000A'    06 05                    LD      B,BYTE+1
000C'    AF                       XOR     A
000D'    CD 0000*                 CALL    .CLR
0010'    11 00A3'                 LD      DE,WRKS
0013'    06 05                    LD      B,BYTE+1
0015'    CD 0000*                 CALL    .CLR
0018'    32 00AD'                 LD      (SGND),A
001B'    2A 00AE'        JTOL3:   LD      HL,(SRCADR)
001E'    7E                       LD      A,(HL)
001F'    23                       INC     HL
0020'    22 00AE'                 LD      (SRCADR),HL
0023'    FE 20                    CP      SPACE
0025'    28 F4                    JR      Z,JTOL3
0027'    FE 2B                    CP      PLUS
0029'    28 0D                    JR      Z,JTOL1
002B'    FE 2D                    CP      MINUS
002D'    28 41                    JR      Z,JTOL2
002F'    CD 0000*                 CALL    .DEC
0032'    DA 009A'                 JP      C,DERR
0035'    32 009E'                 LD      (WRKT),A
0038'    2A 00AE'        JTOL1:   LD      HL,(SRCADR)
003B'    7E                       LD      A,(HL)
003C'    23                       INC     HL
003D'    22 00AE'                 LD      (SRCADR),HL
0040'    CD 0000*                 CALL    .STP
0043'    28 32                    JR      Z,JTOL4
0045'    CD 0000*                 CALL    .DEC
0048'    38 50                    JR      C,DERR
004A'    32 00A3'                 LD      (WRKS),A
004D'    21 009E'                 LD      HL,WRKT
```

```
        0050'   11 00A8'                       LD      DE,WRKD
        0053'   06 05                          LD      B,BYTE+1
        0055'   CD 0000*                       CALL    .M10
        0058'   21 00A3'                       LD      HL,WRKS
        005B'   11 009E'                       LD      DE,WRKT
        005E'   06 05                          LD      B,BYTE+1
        0060'   CD 0000*                       CALL    .ADD
        0063'   3A 00A2'                       LD      A,(WRKT+BYTE)
        0066'   A7                             AND     A
        0067'   28 CF                          JR      Z,JTOL1
        0069'   3E 81                 JOVFL:   LD      A,81H
        006B'   A7                    JTOLR:   AND     A
        006C'   2A 00AE'                       LD      HL,(SRCADR)
        006F'   C9                             RET
        0070'   21 00AD'              JTOL2:   LD      HL,SGND
        0073'   36 80                          LD      (HL),80H
        0075'   18 C1                          JR      JTOL1
        0077'   3A 00AD'              JTOL4:   LD      A,(SGND)
        007A'   A7                             AND     A
        007B'   28 0F                          JR      Z,JTOL5
        007D'   11 009E'                       LD      DE,WRKT
        0080'   06 04                          LD      B,BYTE
        0082'   CD 0000*                       CALL    .NEG
        0085'   21 00A1'                       LD      HL,WRKT+BYTE-
        0088'   CB 7E                          BIT     7,(HL)
        008A'   28 DD                          JR      Z,JOVFL
        008C'   21 009E'              JTOL5:   LD      HL,WRKT
        008F'   ED 5B 00B0'                    LD      DE,(DSTADR)
        0093'   06 04                          LD      B,BYTE
        0095'   ED B0                          LDIR
        0097'   AF                             XOR     A
        0098'   18 D1                          JR      JTOLR
        009A'   3E 01                 DERR:    LD      A,1
        009C'   18 CD                          JR      JTOLR
                                      ;
        009E'                         WRKT:    DEFS    BYTE+1
        00A3'                         WRKS:    DEFS    BYTE+1
        00A8'                         WRKD:    DEFS    BYTE+1
        00AD'   00                    SGND:    DEFB    0
        00AE'   0000                  SRCADR:  DEFW    0
        00B0'   0000                  DSTADR:  DEFW    0
                                      ;
                                               END

Macros:

Symbols:
.ADD    0061*   .CLR    0016*   .DEC    0046*   .M10    0056*
.NEG    0083*   .STP    0041*   BYTE    0004    DERR    009A'
DSTADR  00B0'   JOVFL   0069'   JTOL    0000I'  JTOL1   0038'
JTOL2   0070'   JTOL3   001B'   JTOL4   0077'   JTOL5   008C'
JTOLR   006B'   MINUS   002D    PLUS    002B    SGND    00AD'
SPACE   0020    SRCADR  00AE'   WRKD    00A8'   WRKS    00A3'
WRKT    009E'


No  Fatal error(s)
```

LA

```
LA (V02/L02)    MACRO-80 3.22    10-Apr-79          PAGE    1

                                                TITLE    LA (V02/L02)
                                        ;
                                                GLOBAL   LADD
                                                GLOBAL   LSUB
                                                GLOBAL   LCMP
                                                GLOBAL   LMLT
                                                GLOBAL   LDIV
                                                GLOBAL   LM
                                                GLOBAL   LD
                                                GLOBAL   LMOV
                                                EXTERNAL         .ADD
                                                EXTERNAL         .SUB
                                                EXTERNAL         .ZRC
                                                EXTERNAL         .NEG
                                                EXTERNAL         .CLR
                                                EXTERNAL         .SR1
                                                EXTERNAL         .SL1
                                                EXTERNAL         .CMP
                                        ;
        0004                            BYTE    EQU      4
                                        ;
        0000'   22 029B'                LADD:   LD       (SRCADR),HL
        0003'   ED 53 029D'                     LD       (DSTADR),DE
        0007'   EB                              EX       DE,HL
        0008'   11 028B'                        LD       DE,WRKT
        000B'   01 0004                         LD       BC,BYTE
        000E'   ED B0                           LDIR
        0010'   2A 029B'                        LD       HL,(SRCADR)
        0013'   11 028B'                        LD       DE,WRKT
        0016'   06 04                           LD       B,BYTE
        0018'   CD 0000*                        CALL     .ADD
        001B'   EA 0262'                        JP       PE,OVFL
        001E'   ED 5B 029D'                     LD       DE,(DSTADR)
        0022'   21 028B'                        LD       HL,WRKT
        0025'   01 0004                         LD       BC,BYTE
        0028'   ED B0                           LDIR
        002A'   AF                              XOR      A
        002B'   C9                              RET
                                        ;
        002C'   22 029B'                LSUB:   LD       (SRCADR),HL
        002F'   ED 53 029D'                     LD       (DSTADR),DE
        0033'   EB                              EX       DE,HL
        0034'   11 028B'                        LD       DE,WRKT
        0037'   01 0004                         LD       BC,BYTE
        003A'   ED B0                           LDIR
        003C'   2A 029B'                        LD       HL,(SRCADR)
        003F'   11 028B'                        LD       DE,WRKT
        0042'   06 04                           LD       B,BYTE
        0044'   CD 0000*                        CALL     .SUB
```

```
0047'   EA 0262'                JP      PE,OVFL
004A'   ED 5B 029D'             LD      DE,(DSTADR)
004E'   21 028B'                LD      HL,WRKT
0051'   01 0004                 LD      BC,BYTE
0054'   ED B0                   LDIR
0056'   AF                      XOR     A
0057'   C9                      RET

                        ;
0058'   22 029B'        LCMP:   LD      (SRCADR),HL
005B'   EB                      EX      DE,HL
005C'   11 028B'                LD      DE,WRKT
005F'   01 0004                 LD      BC,BYTE
0062'   ED B0                   LDIR
0064'   11 028B'                LD      DE,WRKT
0067'   2A 029B'                LD      HL,(SRCADR)
006A'   06 04                   LD      B,BYTE
006C'   CD 0000*                CALL    .SUB
006F'   3A 028E'                LD      A,(WRKT+BYTE-1)
0072'   B7                      OR      A
0073'   FA 0079'                JP      M,LCMP1
0076'   28 03                   JR      Z,LCMP2
0078'   C9                      RET

                        ;
0079'   37              LCMP1:  SCF
007A'   C9                      RET
007B'   21 028B'        LCMP2:  LD      HL,WRKT
007E'   06 04                   LD      B,BYTE
0080'   CD 0000*                CALL    .ZRC
0083'   C8                      RET     Z
0084'   3E 7F                   LD      A,7FH
0086'   B7                      OR      A
0087'   C9                      RET

                        ;
0088'   22 029F'        LMLT:   LD      (SRC),HL
008B'   EB                      EX      DE,HL
008C'   ED 53 02A1'             LD      (DST),DE
0090'   11 0293'                LD      DE,TEMP
0093'   01 0004                 LD      BC,BYTE
0096'   ED B0                   LDIR
0098'   2A 029F'                LD      HL,(SRC)
009B'   11 0293'                LD      DE,TEMP
009E'   CD 0107'                CALL    LM
00A1'   21 0296'                LD      HL,TEMP+BYTE-1
00A4'   CB 7E                   BIT     7,(HL)
00A6'   23                      INC     HL
00A7'   06 04                   LD      B,BYTE
00A9'   20 15                   JR      NZ,LMM
00AB'   7E              LM1:    LD      A,(HL)
00AC'   A7                      AND     A
00AD'   C2 0262'                JP      NZ,OVFL
00B0'   23                      INC     HL
00B1'   10 F8                   DJNZ    LM1
00B3'   21 0293'        LM2:    LD      HL,TEMP
00B6'   ED 5B 02A1'             LD      DE,(DST)
00BA'   01 0004                 LD      BC,BYTE
00BD'   ED B0                   LDIR
00BF'   C9                      RET
00C0'   7E              LMM:    LD      A,(HL)
00C1'   FE FF                   CP      0FFH
00C3'   C2 0262'                JP      NZ,OVFL
00C6'   23                      INC     HL
00C7'   10 F7                   DJNZ    LMM
00C9'   18 E8                   JR      LM2

                        ;
00CB'   22 029F'        LDIV:   LD      (SRC),HL
00CE'   EB                      EX      DE,HL
```

```
00CF'   22 02A1'                LD      (DST),HL
00D2'   11 0293'                LD      DE,TEMP
00D5'   01 0004                 LD      BC,BYTE
00D8'   ED B0                   LDIR
00DA'   21 0296'                LD      HL,TEMP+BYTE-1
00DD'   CB 7E                   BIT     7,(HL)
00DF'   23                      INC     HL
00E0'   06 04                   LD      B,BYTE
00E2'   20 1B                   JR      NZ,LDM
00E4'   AF                      XOR     A
00E5'   77              LD1:    LD      (HL),A
00E6'   23                      INC     HL
00E7'   10 FC                   DJNZ    LD1
00E9'   2A 029F'        LD2:    LD      HL,(SRC)
00EC'   11 0293'                LD      DE,TEMP
00EF'   CD 0197'                CALL    LD
00F2'   21 0297'                LD      HL,TEMP+BYTE
00F5'   ED 5B 02A1'             LD      DE,(DST)
00F9'   01 0004                 LD      BC,BYTE
00FC'   ED B0                   LDIR
00FE'   C9                      RET
00FF'   3E FF           LDM:    LD      A,0FFH
0101'   77              LD5:    LD      (HL),A
0102'   23                      INC     HL
0103'   10 FC                   DJNZ    LD5
0105'   18 E2                   JR      LD2
                        ;
0107'   22 029B'        LM:     LD      (SRCADR),HL
010A'   ED 53 029D'             LD      (DSTADR),DE
010E'   11 027F'                LD      DE,WRKS
0111'   01 0004                 LD      BC,BYTE
0114'   ED B0                   LDIR
0116'   1B                      DEC     DE
0117'   1A                      LD      A,(DE)
0118'   E6 80                   AND     80H
011A'   32 02A3'                LD      (SGNS),A
011D'   28 08                   JR      Z,LMLT1
011F'   11 027F'                LD      DE,WRKS
0122'   06 04                   LD      B,BYTE
0124'   CD 0000*                CALL    .NEG
0127'   2A 029D'        LMLT1:  LD      HL,(DSTADR)
012A'   11 0283'                LD      DE,WRKD
012D'   01 0004                 LD      BC,BYTE
0130'   ED B0                   LDIR
0132'   1B                      DEC     DE
0133'   1A                      LD      A,(DE)
0134'   E6 80                   AND     80H
0136'   32 02A5'                LD      (SGND),A
0139'   28 08                   JR      Z,LMLT2
013B'   11 0283'                LD      DE,WRKD
013E'   06 04                   LD      B,BYTE
0140'   CD 0000*                CALL    .NEG
0143'   11 0287'        LMLT2:  LD      DE,WRKD+BYTE
0146'   06 04                   LD      B,BYTE
0148'   AF                      XOR     A
0149'   CD 0000*                CALL    .CLR
014C'   11 028B'                LD      DE,WRKT
014F'   06 08                   LD      B,BYTE*2
0151'   CD 0000*                CALL    .CLR
0154'   06 40                   LD      B,BYTE*8*2
0156'   C5              LMLP:   PUSH    BC
0157'   21 0282'                LD      HL,WRKS+BYTE-1
015A'   06 04                   LD      B,BYTE
015C'   CD 0000*                CALL    .SR1
015F'   30 0B                   JR      NC,LMLT3
0161'   11 028B'                LD      DE,WRKT
```

```
0164'   21 0283'                LD      HL,WRKD
0167'   06 08                   LD      B,BYTE*2
0169'   CD 0000*                CALL    .ADD
016C'   21 0283'        LMLT3:  LD      HL,WRKD
016F'   06 08                   LD      B,BYTE*2
0171'   CD 0000*                CALL    .SL1
0174'   C1                      POP     BC
0175'   10 DF                   DJNZ    LMLP
0177'   3A 02A3'                LD      A,(SGNS)
017A'   4F                      LD      C,A
017B'   3A 02A5'                LD      A,(SGND)
017E'   A9                      XOR     C
017F'   28 08                   JR      Z,LMLT4
0181'   11 028B'                LD      DE,WRKT
0184'   06 08                   LD      B,BYTE*2
0186'   CD 0000*                CALL    .NEG
0189'   21 028B'        LMLT4:  LD      HL,WRKT
018C'   ED 5B 029D'             LD      DE,(DSTADR)
0190'   01 0008                 LD      BC,BYTE*2
0193'   ED B0                   LDIR
0195'   AF                      XOR     A
0196'   C9                      RET
                        ;
0197'   22 029B'        LD:     LD      (SRCADR),HL
019A'   ED 53 029D'             LD      (DSTADR),DE
019E'   11 027F'                LD      DE,WRKS
01A1'   01 0004                 LD      BC,BYTE
01A4'   ED B0                   LDIR
01A6'   1B                      DEC     DE
01A7'   1A                      LD      A,(DE)
01A8'   E6 80                   AND     80H
01AA'   32 02A3'                LD      (SGNS),A
01AD'   28 08                   JR      Z,LDIV1
01AF'   11 027F'                LD      DE,WRKS
01B2'   06 04                   LD      B,BYTE
01B4'   CD 0000*                CALL    .NEG
01B7'   2A 029D'        LDIV1:  LD      HL,(DSTADR)
01BA'   11 0283'                LD      DE,WRKD
01BD'   01 0008                 LD      BC,BYTE*2
01C0'   ED B0                   LDIR
01C2'   1B                      DEC     DE
01C3'   1A                      LD      A,(DE)
01C4'   E6 80                   AND     80H
01C6'   32 02A5'                LD      (SGND),A
01C9'   28 08                   JR      Z,LDIV2
01CB'   11 0283'                LD      DE,WRKD
01CE'   06 08                   LD      B,BYTE*2
01D0'   CD 0000*                CALL    .NEG
01D3'   11 028A'        LDIV2:  LD      DE,WRKD+BYTE*2-
01D6'   21 0282'                LD      HL,WRKS+BYTE-1
01D9'   06 04                   LD      B,BYTE
01DB'   CD 0000*                CALL    .CMP
01DE'   D2 0262'                JP      NC,OVFL
01E1'   11 028B'                LD      DE,WRKT
01E4'   06 04                   LD      B,BYTE
01E6'   AF                      XOR     A
01E7'   CD 0000*                CALL    .CLR
01EA'   06 21                   LD      B,BYTE*8+1
01EC'   C5              LDLP:   PUSH    BC
01ED'   06 04                   LD      B,BYTE
01EF'   21 028B'                LD      HL,WRKT
01F2'   CD 0000*                CALL    .SL1
01F5'   11 028A'                LD      DE,WRKD+BYTE*2-1
01F8'   21 0282'                LD      HL,WRKS+BYTE-1
01FB'   06 04                   LD      B,BYTE
01FD'   CD 0000*                CALL    .CMP
```

```
0200'   38 10                       JR      C,LDIV3
0202'   21 028B'                    LD      HL,WRKT
0205'   CB C6                       SET     0,(HL)
0207'   11 0287'                    LD      DE,WRKD+BYTE
020A'   21 027F'                    LD      HL,WRKS
020D'   06 04                       LD      B,BYTE
020F'   CD 0000*                    CALL    .SUB
0212'   21 0283'           LDIV3:   LD      HL,WRKD
0215'   06 08                       LD      B,BYTE*2
0217'   CD 0000*                    CALL    .SL1
021A'   C1                          POP     BC
021B'   10 CF                       DJNZ    LDLP
021D'   21 028A'                    LD      HL,WRKD+BYTE*2-1
0220'   06 04                       LD      B,BYTE
0222'   CD 0000*                    CALL    .SR1
0225'   3A 028E'                    LD      A,(WRKT+BYTE-1)
0228'   E6 80                       AND     80H
022A'   20 36                       JR      NZ,OVFL
022C'   3A 02A5'                    LD      A,(SGND)
022F'   A7                          AND     A
0230'   28 08                       JR      Z,LDIV4
0232'   11 0287'                    LD      DE,WRKD+BYTE
0235'   06 04                       LD      B,BYTE
0237'   CD 0000*                    CALL    .NEG
023A'   3A 02A5'           LDIV4:   LD      A,(SGND)
023D'   4F                          LD      C,A
023E'   3A 02A3'                    LD      A,(SGNS)
0241'   A9                          XOR     C
0242'   28 08                       JR      Z,LDIV5
0244'   11 028B'                    LD      DE,WRKT
0247'   06 04                       LD      B,BYTE
0249'   CD 0000*                    CALL    .NEG
024C'   ED 5B 029D'        LDIV5:   LD      DE,(DSTADR)
0250'   21 0287'                    LD      HL,WRKD+BYTE
0253'   01 0004                     LD      BC,BYTE
0256'   ED B0                       LDIR
0258'   21 028B'                    LD      HL,WRKT
025B'   01 0004                     LD      BC,BYTE
025E'   ED B0                       LDIR
0260'   AF                          XOR     A
0261'   C9                          RET
0262'   3E 81              OVFL:    LD      A,81H
0264'   A7                          AND     A
0265'   C9                          RET
                           ;
0266'   22 029B'           LMOV:    LD      (SRCADR),HL
0269'   01 0004                     LD      BC,BYTE
026C'   ED B0                       LDIR
026E'   2B                          DEC     HL
026F'   7E                          LD      A,(HL)
0270'   B7                          OR      A
0271'   C0                          RET     NZ
0272'   2A 029B'                    LD      HL,(SRCADR)
0275'   06 04                       LD      B,BYTE
0277'   CD 0000*                    CALL    .ZRC
027A'   C8                          RET     Z
027B'   3E 7F                       LD      A,7FH
027D'   B7                          OR      A
027E'   C9                          RET
                           ;
027F'                      WRKS:    DEFS    BYTE
0283'                      WRKD:    DEFS    BYTE*2
028B'                      WRKT:    DEFS    BYTE*2
0293'                      TEMP:    DEFS    BYTE*2
029B'   0000               SRCADR:  DEFW    0
029D'   0000               DSTADR:  DEFW    0
```

```
            029F'   0000                                SRC:    DEFW    0
            02A1'   0000                                DST:    DEFW    0
            02A3'   0000                                SGNS:   DEFW    0
            02A5'   0000                                SGND:   DEFW    0
                                                        ;
                                                                END
Macros:

Symbols:
.ADD     016A*    .CLR     01E8*    .CMP     01FE*    .NEG     024A*
.SL1     0218*    .SR1     0223*    .SUB     0210*    .ZRC     0278*
BYTE     0004     DST      02A1'    DSTADR   029D'    LADD     0000I'
LCMP     0058I'   LCMP1    0079'    LCMP2    007B'    LD       0197I'
LD1      00E5'    LD2      00E9'    LD5      0101'    LDIV     00CBI'
LDIV1    01B7'    LDIV2    01D3'    LDIV3    0212'    LDIV4    023A'
LDIV5    024C'    LDLP     01EC'    LDM      00FF'    LM       0107I'
LM1      00AB'    LM2      00B3'    LMLP     0156'    LMLT     0088I'
LMLT1    0127'    LMLT2    0143'    LMLT3    016C'    LMLT4    0189'
LMM      00C0'    LMOV     0266I'   LSUB     002CI'   OVFL     0262'
SGND     02A5'    SGNS     02A3'    SRC      029F'    SRCADR   029B'
TEMP     0293'    WRKD     0283'    WRKS     027F'    WRKT     028B'


No  Fatal error(s)
```

LABS

```
LABS (V01/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1

                                                TITLE   LABS (V01/L01)
                                        ;
                                                GLOBAL  LABS
                                                EXTERNAL                .NEG
                                        ;
        0000'                           LABS    EQU     $
        0000'   01 0004                         LD      BC,4
        0003'   ED B0                           LDIR
        0005'   1B                              DEC     DE
        0006'   EB                              EX      DE,HL
        0007'   CB 7E                           BIT     7,(HL)
        0009'   28 09                           JR      Z,LABS1
        000B'   2B                              DEC     HL
        000C'   2B                              DEC     HL
        000D'   2B                              DEC     HL
        000E'   EB                              EX      DE,HL
        000F'   06 04                           LD      B,4
        0011'   CD 0000*                        CALL    .NEG
        0014'   AF                      LABS1:  XOR     A
        0015'   C9                              RET
                                        ;
                                                END

Macros:

Symbols:
.NEG    0012*   LABS    0000I'  LABS1   0014'


No  Fatal error(s)
```

LDFIX

```
LDFIX (V02/L01)              MACRO-80 3.22   10-Apr-79       PAGE    1

                                                TITLE   LDFIX (V02/L01
                                        ;
                                                GLOBAL  LDFIX
                                                EXTERNAL        .CLR
                                                EXTERNAL        .SL4
                                                EXTERNAL        .NEG
                                        ;
        0004                            BYTEL   EQU     4
        0008                            BYTEF   EQU     8
                                        ;
        0000'                           LDFIX   EQU     $
        0000'   22 0099'                        LD      (SRCADR),HL
        0003'   ED 53 009B'                     LD      (DSTADR),DE
        0007'   7D                              LD      A,L
        0008'   C6 07                           ADD     A,BYTEF-1
        000A'   6F                              LD      L,A
        000B'   30 01                           JR      NC,LFIX1
        000D'   24                              INC     H
        000E'   7E                      LFIX1:  LD      A,(HL)
        000F'   E6 7F                           AND     7FH
        0011'   FE 41                           CP      41H
        0013'   DA 0083'                        JP      C,LFIX2
        0016'   FE 49                           CP      41H+BYTEL*2
        0018'   D2 008A'                        JP      NC,OVFL
        001B'   D6 40                           SUB     40H
        001D'   32 009E'                        LD      (EXPD),A
        0020'   11 008E'                        LD      DE,WRKT
        0023'   06 04                           LD      B,BYTEL
        0025'   AF                              XOR     A
        0026'   CD 0000*                        CALL    .CLR
        0029'   11 0092'                        LD      DE,WRKD
        002C'   2A 0099'                        LD      HL,(SRCADR)
        002F'   01 0007                         LD      BC,BYTEF-1
        0032'   ED B0                           LDIR
        0034'   3A 009E'                        LD      A,(EXPD)
        0037'   47                              LD      B,A
        0038'   C5                      LFIXLP: PUSH    BC
        0039'   21 008E'                        LD      HL,WRKT
        003C'   06 04                           LD      B,BYTEL
        003E'   CD 0000*                        CALL    .SL4
        0041'   3A 0098'                        LD      A,(WRKD+BYT
        0044'   0F                              RRCA
        0045'   0F                              RRCA
        0046'   0F                              RRCA
        0047'   0F                              RRCA
        0048'   E6 0F                           AND     0FH
        004A'   21 008E'                        LD      HL,WRKT
        004D'   AE                              XOR     (HL)
        004E'   77                              LD      (HL),A
```

```
004F'   21 0092'                    LD      HL,WRKD
0052'   06 07                       LD      B,BYTEF-1
0054'   CD 0000*                    CALL    .SL4
0057'   C1                          POP     BC
0058'   10 DE                       DJNZ    LFIXLP
005A'   3A 009D'                    LD      A,(SGND)
005D'   A7                          AND     A
005E'   28 08                       JR      Z,LFIX3
0060'   11 008E'                    LD      DE,WRKT
0063'   06 04                       LD      B,BYTEL
0065'   CD 0000*                    CALL    .NEG
0068'   3A 0091'          LFIX3:    LD      A,(WRKT+BYTEL-1)
006B'   E6 80                       AND     80H
006D'   4F                          LD      C,A
006E'   3A 009D'                    LD      A,(SGND)
0071'   A9                          XOR     C
0072'   C2 008A'                    JP      NZ,OVFL
0075'   ED 5B 009B'                 LD      DE,(DSTADR)
0079'   21 008E'                    LD      HL,WRKT
007C'   01 0004                     LD      BC,BYTEL
007F'   ED B0                       LDIR
0081'   AF                          XOR     A
0082'   C9                          RET
0083'   06 04             LFIX2:    LD      B,BYTEL
0085'   AF                          XOR     A
0086'   CD 0000*                    CALL    .CLR
0089'   C9                          RET
008A'   3E 81             OVFL:     LD      A,81H
008C'   A7                          AND     A
008D'   C9                          RET

                          ;
008E'                     WRKT:     DEFS    BYTEL
0092'                     WRKD:     DEFS    BYTEF-1
0099'   0000              SRCADR:   DEFW    0
009B'   0000              DSTADR:   DEFW    0
009D'   00                SGND:     DEFB    0
009E'   00                EXPD:     DEFB    0
                          ;
                                    END

Macros:

Symbols:
.CLR    0087*    .NEG    0066*    .SL4    0055*    BYTEF   0008
BYTEL   0004     DSTADR  009B'    EXPD    009E'    LDFIX   0000I'
LFIX1   000E'    LFIX2   0083'    LFIX3   0068'    LFIXLP  0038'
OVFL    008A'    SGND    009D'    SRCADR  0099'    WRKD    0092'
WRKT    008E'


No  Fatal error(s)
```

LMOD

```
LMOD (V01/L01)           MACRO-80 3.22    10-Apr-79         PAGE    1

                                          TITLE    LMOD (V01/L01)
                                  ;
                                          GLOBAL   LMOD
                                          EXTERNAL         LD
                                  ;
        0000'                     LMOD    EQU      $
        0000'    22 003C'                 LD       (SRCADR),HL
        0003'    EB                       EX       DE,HL
        0004'    22 003E'                 LD       (DSTADR),HL
        0007'    11 0040'                 LD       DE,TEMP
        000A'    01 0004                  LD       BC,4
        000D'    ED B0                    LDIR
        000F'    21 0043'                 LD       HL,TEMP+3
        0012'    CB 7E                    BIT      7,(HL)
        0014'    23                       INC      HL
        0015'    06 04                    LD       B,4
        0017'    20 1B                    JR       NZ,LMOD1
        0019'    AF                       XOR      A
        001A'    77               LMOD2:  LD       (HL),A
        001B'    23                       INC      HL
        001C'    10 FC                    DJNZ     LMOD2
        001E'    2A 003C'         LMOD3:  LD       HL,(SRCADR)
        0021'    11 0040'                 LD       DE,TEMP
        0024'    CD 0000*                 CALL     LD
        0027'    21 0040'                 LD       HL,TEMP
        002A'    ED 5B 003E'              LD       DE,(DSTADR)
        002E'    01 0004                  LD       BC,4
        0031'    ED B0                    LDIR
        0033'    C9                       RET
        0034'    3E FF            LMOD1:  LD       A,0FFH
        0036'    77               LMOD6:  LD       (HL),A
        0037'    23                       INC      HL
        0038'    10 FC                    DJNZ     LMOD6
        003A'    18 E2                    JR       LMOD3
                                  ;
        003C'    0000             SRCADR: DEFW     0
        003E'    0000             DSTADR: DEFW     0
        0040'    00 00 00 00      TEMP:   DEFB     0,0,0,0,0,0,0,0
        0044'    00 00 00 00
                                  ;
                                          END

Macros:

Symbols:
DSTADR  003E'   LD      0025*   LMOD    0000I'  LMOD1   0034'
LMOD2   001A'   LMOD3   001E'   LMOD6   0036'   SRCADR  003C'
TEMP    0040'

  No  Fatal error(s)
```

LSGN

```
LSGN (V01/L01)          MACRO-80 3.22   10-Apr-79       PAGE    1


                                          TITLE   LSGN (V01/L01
                                  ;
                                          GLOBAL  LSGN
                                  ;
        0000'                     LSGN    EQU     $
        0000'   23                        INC     HL
        0001'   23                        INC     HL
        0002'   23                        INC     HL
        0003'   CB 7E                     BIT     7,(HL)
        0005'   20 17                     JR      NZ,LSGN1
        0007'   06 04                     LD      B,4
        0009'   AF                        XOR     A
        000A'   B6                LSGN2:  OR      (HL)
        000B'   2B                        DEC     HL
        000C'   10 FC                     DJNZ    LSGN2
        000E'   06 04                     LD      B,4
        0010'   28 06                     JR      Z,LSGN4
        0012'   05                        DEC     B
        0013'   3E 01                     LD      A,1
        0015'   12                        LD      (DE),A
        0016'   13                        INC     DE
        0017'   AF                        XOR     A
        0018'   12                LSGN4:  LD      (DE),A
        0019'   13                        INC     DE
        001A'   10 FC                     DJNZ    LSGN4
        001C'   AF                        XOR     A
        001D'   C9                        RET
        001E'   3E FF             LSGN1:  LD      A,0FFH
        0020'   06 04                     LD      B,4
        0022'   18 F4                     JR      LSGN4
                                  ;
                                          END


Macros:

Symbols:
LSGN    0000I'  LSGN1   001E'   LSGN2   000A'   LSGN4   0018'



No  Fatal error(s)
```

LTOJ

```
LTOJ (V02/L01)  MACRO-80 3.22   10-Apr-79         PAGE    1

                                             TITLE    LTOJ (V02/L01)
                                     ;
                                             GLOBAL   LTOJ
                                             EXTERNAL          .CLR
                                             EXTERNAL          .NEG
                                             EXTERNAL          .D10
                                             EXTERNAL          .ZRC
                                     ;
        0020                         SPACE   EQU      20H
        002D                         MINUS   EQU      2DH
        0030                         NBASE   EQU      30H
        0004                         BYTE    EQU      4
        000B                         BYTEC   EQU      BYTE*8*3/10+2
                                     ;
        0000'                        LTOJ    EQU      $
        0000'   ED 53 006E'                  LD       (DSTADR),DE
        0004'   11 0061'                     LD       DE,WRKT
        0007'   01 0004                      LD       BC,BYTE
        000A'   ED B0                        LDIR
        000C'   2B                           DEC      HL
        000D'   7E                           LD       A,(HL)
        000E'   E6 80                        AND      80H
        0010'   32 006B'                     LD       (SGND),A
        0013'   28 08                        JR       Z,LTOJ1
        0015'   11 0061'                     LD       DE,WRKT
        0018'   06 04                        LD       B,BYTE
        001A'   CD 0000*                     CALL     .NEG
        001D'   ED 5B 006E'          LTOJ1:  LD       DE,(DSTADR)
        0021'   06 0B                        LD       B,BYTEC
        0023'   3E 20                        LD       A,SPACE
        0025'   CD 0000*                     CALL     .CLR
        0028'   ED 53 006C'                  LD       (SRCADR),DE
        002C'   1B                           DEC      DE
        002D'   ED 53 006E'                  LD       (DSTADR),DE
        0031'   21 0061'             LTOJ2:  LD       HL,WRKT
        0034'   11 0066'                     LD       DE,WRKD
        0037'   06 04                        LD       B,BYTE
        0039'   CD 0000*                     CALL     .D10
        003C'   EE 30                        XOR      NBASE
        003E'   2A 006E'                     LD       HL,(DSTADR)
        0041'   77                           LD       (HL),A
        0042'   2B                           DEC      HL
        0043'   22 006E'                     LD       (DSTADR),HL
        0046'   21 0061'                     LD       HL,WRKT
        0049'   06 04                        LD       B,BYTE
        004B'   CD 0000*                     CALL     .ZRC
        004E'   20 E1                        JR       NZ,LTOJ2
        0050'   3A 006B'                     LD       A,(SGND)
        0053'   A7                           AND      A
```

```
0054'  28 05                      JR     Z,LTOJ3
0056'  2A 006E'                   LD     HL,(DSTADR)
0059'  36 2D                      LD     (HL),MINUS
005B'  AF                LTOJ3:   XOR    A
005C'  ED 5B 006C'                LD     DE,(SRCADR)
0060'  C9                         RET
                         ;
0061'                    WRKT:    DEFS   BYTE+1
0066'                    WRKD:    DEFS   BYTE+1
006B'  00                SGND:    DEFB   0
006C'  0000              SRCADR:  DEFW   0
006E'  0000              DSTADR:  DEFW   0
                         ;
                                  END
```

```
Macros:

Symbols:
.CLR    0026*   .D10    003A*   .NEG    001B*   .ZRC    004C*
BYTE    0004    BYTEC   000B    DSTADR  006E'   LTOJ    0000I'
LTOJ1   001D'   LTOJ2   0031'   LTOJ3   005B'   MINUS   002D
NBASE   0030    SGND    006B'   SPACE   0020    SRCADR  006C'
WRKD    0066'   WRKT    0061'


No  Fatal error(s)
```

ROUND

```
ROUND (V02/L02) MACRO-80 3.22   10-Apr-79        PAGE    1

                                        TITLE   ROUND (V02/L02)
                                ;
                                        GLOBAL  .RND
                                ;
     0000'                      .RND    EQU     $
     0000'   1A                         LD      A,(DE)
     0001'   A7                         AND     A
     0002'   F0                         RET     P
     0003'   E6 7F                      AND     7FH
     0005'   12                         LD      (DE),A
     0006'   13                         INC     DE
     0007'   0E 00                      LD      C,0
     0009'   37                         SCF
     000A'   1A                 RND1:   LD      A,(DE)
     000B'   89                         ADC     A,C
     000C'   12                         LD      (DE),A
     000D'   13                         INC     DE
     000E'   10 FA                      DJNZ    RND1
     0010'   C9                         RET

                                        END
Macros:

Symbols:
 RND     0000I'  RND1    000A'


No  Fatal error(s)
```

```
        MACRO-80 3.3    09-MAY-79       PAGE    1
                                00100   .Z80
                                00200   ;       NUMERIC LIST OF INSTRUCTION SET.1
                                00300
0012                            00400   X       EQU     012H
0034                            00500   Y       EQU     034H
1234                            00600   XY      EQU     01234H
0000'   00                      00700           NOP
0001'   01 1234                 00800           LD      BC,XY
0004'   02                      00900           LD      (BC),A
0005'   03                      01000           INC     BC
0006'   04                      01100           INC     B
0007'   05                      01200           DEC     B
0008'   06 12                   01300           LD      B,X
000A'   07                      01400           RLCA
000B'   08                      01500           EX      AF,AF'
000C'   09                      01600           ADD     HL,BC
000D'   0A                      01700           LD      A,(BC)
000E'   0B                      01800           DEC     BC
000F'   0C                      01900           INC     C
0010'   0D                      02000           DEC     C
0011'   0E 12                   02100           LD      C,X
0013'   0F                      02200           RRCA
0014'   10 03                   02300           DJNZ    SYM1
0016'   11 1234                 02400           LD      DE,XY
0019'   12                      02500   SYM1:   LD      (DE),A
001A'   13                      02600           INC     DE
001B'   14                      02700           INC     D
001C'   15                      02800           DEC     D
001D'   16 12                   02900           LD      D,X
001F'   17                      03000           RLA
0020'   18 01                   03100           JR      SYM2
0022'   19                      03200           ADD     HL,DE
0023'   1A                      03300   SYM2:   LD      A,(DE)
0024'   1B                      03400           DEC     DE
0025'   1C                      03500           INC     E
0026'   1D                      03600           DEC     E
0027'   1E 12                   03700           LD      E,X
0029'   1F                      03800           RRA
002A'   20 03                   03900           JR      NZ,SYM3
002C'   21 1234                 04000           LD      HL,XY
002F'   22 1234                 04100   SYM3:   LD      (XY),HL
0032'   23                      04200           INC     HL
0033'   24                      04300           INC     H
0034'   25                      04400           DEC     H
0035'   26 12                   04500           LD      H,X
0037'   27                      04600           DAA
```

```
      MACRO-80 3.3    09-MAY-79        PAGE    1-1
0038'    28 01                 04700            JR      Z,SYM4
003A'    29                    04800            ADD     HL,HL
003B'    2A 1234               04900    SYM4:   LD      HL,(XY)
003E'    2B                    05000            DEC     HL
003F'    2C                    05100            INC     L
0040'    2D                    05200            DEC     L
0041'    2E 12                 05300            LD      L,X
0043'    2F                    05400            CPL
0044'    30 03                 05500            JR      NC,SYM5
0046'    31 1234               05600            LD      SP,XY
0049'    32 1234               05700    SYM5:   LD      (XY),A
004C'    33                    05800            INC     SP
004D'    34                    05900            INC     (HL)
004E'    35                    06000            DEC     (HL)
004F'    36 12                 06100            LD      (HL),X
0051'    37                    06200            SCF
0052'    38 01                 06300            JR      C,SYM6
0054'    39                    06400            ADD     HL,SP
0055'    3A 1234               06500    SYM6.   LD      A,(XY)
0058'    3B                    06600            DEC     SP
0059'    3C                    06700            INC     A
005A'    3D                    06800            DEC     A
005B'    3E 12                 06900            LD      A,X
005D'    3F                    07000            CCF
005E'    40                    07100            LD      B,B
005F'    41                    07200            LD      B,C
0060'    42                    07300            LD      B,D
0061'    43                    07400            LD      B,E
0062'    44                    07500            LD      B,H
0063'    45                    07600            LD      B,L
0064'    46                    07700            LD      B,(HL)
0065'    47                    07800            LD      B,A
0066'    48                    07900            LD      C,B
0067'    49                    08000            LD      C,C
0068'    4A                    08100            LD      C,D
0069'    4B                    08200            LD      C,E
006A'    4C                    08300            LD      C,H
006B'    4D                    08400            LD      C,L
006C'    4E                    08500            LD      C,(HL)
006D'    4F                    08600            LD      C,A
006E'    50                    08700            LD      D,B
006F'    51                    08800            LD      D,C
0070'    52                    08900            LD      D,D
0071'    53                    09000            LD      D,E
0072'    54                    09100            LD      D,H
0073'    55                    09200            LD      D,L
```

```
          MACRO-80 3.3    09-MAY-79     PAGE    1-2
0074'    56                   09300          LD        D,(HL)
0075'    57                   09400          LD        D,A
0076'    58                   09500          LD        E,B
0077'    59                   09600          LD        E,C
0078'    5A                   09700          LD        E,D
0079'    5B                   09800          LD        E,E
007A'    5C                   09900          LD        E,H
007B'    5D                   10000          LD        E,L
007C'    5E                   10100          LD        E,(HL)
007D'    5F                   10200          LD        E,A
007E'    60                   10300          LD        H,B
007F'    61                   10400          LD        H,C
0080'    62                   10500          LD        H,D
0081'    63                   10600          LD        H,E
0082'    64                   10700          LD        H,H
0083'    65                   10800          LD        H,L
0084'    66                   10900          LD        H,(HL)
0085'    67                   11000          LD        H,A
0086'    68                   11100          LD        L,B
0087'    69                   11200          LD        L,C
0088'    6A                   11300          LD        L,D
0089'    6B                   11400          LD        L,E
008A'    6C                   11500          LD        L,H
008B'    6D                   11600          LD        L,L
008C'    6E                   11700          LD        L,(HL)
008D'    6F                   11800          LD        L,A
008E'    70                   11900          LD        (HL),B
008F'    71                   12000          LD        (HL),C
0090'    72                   12100          LD        (HL),D
0091'    73                   12200          LD        (HL),E
0092'    74                   12300          LD        (HL),H
0093'    75                   12400          LD        (HL),L
0094'    76                   12500          HALT
0095'    77                   12600          LD        (HL),A
0096'    78                   12700          LD        A,B
0097'    79                   12800          LD        A,C
0098'    7A                   12900          LD        A,D
0099'    7B                   13000          LD        A,E
009A'    7C                   13100          LD        A,H
009B'    7D                   13200          LD        A,L
009C'    7E                   13300          LD        A,(HL)
009D'    7F                   13400          LD        A,A
009E'    80                   13500          ADD       A,B
009F'    81                   13600          ADD       A,C
00A0'    82                   13700          ADD       A,D
00A1'    83                   13800          ADD       A,E
```

```
        MACRO-80 3.3    09-MAY-79       PAGE    1-3
00A2'   84                      13900           ADD     A,H
00A3'   85                      14000           ADD     A,L
00A4'   86                      14100           ADD     A,(HL)
00A5'   87                      14200           ADD     A,A
00A6'   88                      14300           ADC     A,B
00A7'   89                      14400           ADC     A,C
00A8'   8A                      14500           ADC     A,D
00A9'   8B                      14600           ADC     A,E
00AA'   8C                      14700           ADC     A,H
00AB'   8D                      14800           ADC     A,L
00AC'   8E                      14900           ADC     A,(HL)
00AD'   8F                      15000           ADC     A,A
00AE'   90                      15100           SUB     B
00AF'   91                      15200           SUB     C
00B0'   92                      15300           SUB     D
00B1'   93                      15400           SUB     E
00B2'   94                      15500           SUB     H
00B3'   95                      15600           SUB     L
00B4'   96                      15700           SUB     (HL)
00B5'   97                      15800           SUB     A
00B6'   98                      15900           SBC     A,B
00B7'   99                      16000           SBC     A,C
00B8'   9A                      16100           SBC     A,D
00B9'   9B                      16200           SBC     A,E
00BA'   9C                      16300           SBC     A,H
00BB'   9D                      16400           SBC     A,L
00BC'   9E                      16500           SBC     A,(HL)
00BD'   9F                      16600           SBC     A,A
00BE'   A0                      16700           AND     B
00BF'   A1                      16800           AND     C
00C0'   A2                      16900           AND     D
00C1'   A3                      17000           AND     E
00C2'   A4                      17100           AND     H
00C3'   A5                      17200           AND     L
00C4'   A6                      17300           AND     (HL)
00C5'   A7                      17400           AND     A
00C6'   A8                      17500           XOR     B
00C7'   A9                      17600           XOR     C
00C8'   AA                      17700           XOR     D
00C9'   AB                      17800           XOR     E
00CA'   AC                      17900           XOR     H
00CB'   AD                      18000           XOR     L
00CC'   AE                      18100           XOR     (HL)
00CD'   AF                      18200           XOR     A
                                18300           END
```

```
      MACRO-80 3.3    09-MAY-79      PAGE    S
MACROS.
SYMBOLS.
SYM1    0019'   SYM2    0023'   SYM3    002F'   SYM4    003B'
SYM5    0049'   SYM6    0055'   X       0012    XY      1234
Y       0034
NO  FATAL ERROR(S)

     MACRO-80 3.3    09-MAY-79      PAGE    1
                              00100   .Z80
                              00200   ;  NUMERIC LIST OF INSTRUCTION SET.2
                              00300
0012                          00400   X       EQU     012H
0034                          00500   Y       EQU     034H
1234                          00600   XY      EQU     01234H
0000'   B0                    00700           OR      B
0001'   B1                    00800           OR      C
0002'   B2                    00900           OR      D
0003'   B3                    01000           OR      E
0004'   B4                    01100           OR      H
0005'   B5                    01200           OR      L
0006'   B6                    01300           OR      (HL)
0007'   B7                    01400           OR      A
0008'   B8                    01500           CP      B
0009'   B9                    01600           CP      C
000A'   BA                    01700           CP      D
000B'   BB                    01800           CP      E
000C'   BC                    01900           CP      H
000D'   BD                    02000           CP      L
000E'   BE                    02100           CP      (HL)
000F'   BF                    02200           CP      A
0010'   C0                    02300           RET     NZ
0011'   C1                    02400           POP     BC
0012'   C2 1234               02500           JP      NZ,XY
0015'   C3 1234               02600           JP      XY
0018'   C4 1234               02700           CALL    NZ,XY
001B'   C5                    02800           PUSH    BC
001C'   C6 12                 02900           ADD     A,X
001E'   C7                    03000           RST     0
001F'   C8                    03100           RET     Z
0020'   C9                    03200           RET
0021'   CA 1234               03300           JP      Z,XY
0024'   CB 00                 03400           RLC     B
0026'   CB 01                 03500           RLC     C
0028'   CB 02                 03600           RLC     D
002A'   CB 03                 03700           RLC     E
002C'   CB 04                 03800           RLC     H
002E'   CB 05                 03900           RLC     L
0030'   CB 06                 04000           RLC     (HL)
0032'   CB 07                 04100           RLC     A
                              04200           END
```

```
          MACRO-80 3.3    09-MAY-79        PAGE    S
MACROS:
SYMBOLS:
X        0012    XY       1234    Y        0034
NO  FATAL ERROR(S)

        MACRO-80 3.3    09-MAY-79        PAGE    1
                                00100  .Z80
                                00200  ;    NUMERIC LIST OF INSTRUCTION SET.3
                                00300  ;
0000'   CB 08                   00400         RRC     B
0002'   CB 09                   00500         RRC     C
0004'   CB 0A                   00600         RRC     D
0006'   CB 0B                   00700         RRC     E
0008'   CB 0C                   00800         RRC     H
000A'   CB 0D                   00900         RRC     L
000C'   CB 0E                   01000         RRC     (HL)
000E'   CB 0F                   01100         RRC     A
0010'   CB 10                   01200         RL      B
0012'   CB 11                   01300         RL      C
0014'   CB 12                   01400         RL      D
0016'   CB 13                   01500         RL      E
0018'   CB 14                   01600         RL      H
001A'   CB 15                   01700         RL      L
001C'   CB 16                   01800         RL      (HL)
001E'   CB 17                   01900         RL      A
0020'   CB 18                   02000         RR      B
0022'   CB 19                   02100         RR      C
0024'   CB 1A                   02200         RR      D
0026'   CB 1B                   02300         RR      E
0028'   CB 1C                   02400         RR      H
002A'   CB 1D                   02500         RR      L
002C'   CB 1E                   02600         RR      (HL)
002E'   CB 1F                   02700         RR      A
0030'   CB 20                   02800         SLA     B
0032'   CB 21                   02900         SLA     C
0034'   CB 22                   03000         SLA     D
0036'   CB 23                   03100         SLA     E
0038'   CB 24                   03200         SLA     H
003A'   CB 25                   03300         SLA     L
003C'   CB 26                   03400         SLA     (HL)
003E'   CB 27                   03500         SLA     A
0040'   CB 28                   03600         SRA     B
0042'   CB 29                   03700         SRA     C
0044'   CB 2A                   03800         SRA     D
0046'   CB 2B                   03900         SRA     E
0048'   CB 2C                   04000         SRA     H
004A'   CB 2D                   04100         SRA     L
004C'   CB 2E                   04200         SRA     (HL)
004E'   CB 2F                   04300         SRA     A
                                04400         END
```

```
          MACRO-80 3.3    09-MAY-79       PAGE    S
MACROS:
SYMBOLS:
NO  FATAL ERROR(S)


       MACRO-80 3.3    09-MAY-79       PAGE    1
                               00100   .Z80
                               00200   ;    NUMERIC LIST OF INSTRUCTION SET.4
                               00300   ;
0000'   CB 38                  00310        SRL    B
0002'   CB 39                  00320        SRL    C
0004'   CB 3A                  00330        SRL    D
0006'   CB 3B                  00340        SRL    E
0008'   CB 3C                  00350        SRL    H
000A'   CB 3D                  00360        SRL    L
000C'   CB 3E                  00370        SRL    (HL)
000E'   CB 3F                  00380        SRL    A
0010'   CB 40                  00400        BIT    0,B
0012'   CB 41                  00500        BIT    0,C
0014'   CB 42                  00600        BIT    0,D
0016'   CB 43                  00700        BIT    0,E
0018'   CB 44                  00800        BIT    0,H
001A'   CB 45                  00900        BIT    0,L
001C'   CB 46                  01000        BIT    0,(HL)
001E'   CB 47                  01100        BIT    0,A
0020'   CB 48                  01200        BIT    1,B
0022'   CB 49                  01300        BIT    1,C
0024'   CB 4A                  01400        BIT    1,D
0026'   CB 4B                  01500        BIT    1,E
0028'   CB 4C                  01600        BIT    1,H
002A'   CB 4D                  01700        BIT    1,L
002C'   CB 4E                  01800        BIT    1,(HL)
002E'   CB 4F                  01900        BIT    1,A
0030'   CB 50                  02000        BIT    2,B
0032'   CB 51                  02100        BIT    2,C
0034'   CB 52                  02200        BIT    2,D
0036'   CB 53                  02300        BIT    2,E
0038'   CB 54                  02400        BIT    2,H
003A'   CB 55                  02500        BIT    2,L
003C'   CB 56                  02600        BIT    2,(HL)
003E'   CB 57                  02700        BIT    2,A
0040'   CB 58                  02800        BIT    3,B
0042'   CB 59                  02900        BIT    3,C
0044'   CB 5A                  03000        BIT    3,D
0046'   CB 5B                  03100        BIT    3,E
0048'   CB 5C                  03200        BIT    3,H
004A'   CB 5D                  03300        BIT    3,L
004C'   CB 5E                  03400        BIT    3,(HL)
004E'   CB 5F                  03500        BIT    3,A
                               03600        END
```

```
          MACRO-80 3.3    09-MAY-79        PAGE    S
MACROS.
SYMBOLS.
NO  FATAL ERROR(S)

          MACRO-80 3.3    09-MAY-79        PAGE    1
                                  .Z80
                                  ;   NUMERIC LIST OF INSTRUCTION SET.5
                                  ;
 0000'    CB 60                           BIT     4,B
 0002'    CB 61                           BIT     4,C
 0004'    CB 62                           BIT     4,D
 0006'    CB 63                           BIT     4,E
 0008'    CB 64                           BIT     4,H
 000A'    CB 65                           BIT     4,L
 000C'    CB 66                           BIT     4,(HL)
 000E'    CB 67                           BIT     4,A
 0010'    CB 68                           BIT     5,B
 0012'    CB 69                           BIT     5,C
 0014'    CB 6A                           BIT     5,D
 0016'    CB 6B                           BIT     5,E
 0018'    CB 6C                           BIT     5,H
 001A'    CB 6D                           BIT     5,L
 001C'    CB 6E                           BIT     5,(HL)
 001E'    CB 6F                           BIT     5,A
 0020'    CB 70                           BIT     6,B
 0022'    CB 71                           BIT     6,C
 0024'    CB 72                           BIT     6,D
 0026'    CB 73                           BIT     6,E
 0028'    CB 74                           BIT     6,H
 002A'    CB 75                           BIT     6,L
 002C'    CB 76                           BIT     6,(HL)
 002E'    CB 77                           BIT     6,A
 0030'    CB 78                           BIT     7,B
 0032'    CB 79                           BIT     7,C
 0034'    CB 7A                           BIT     7,D
 0036'    CB 7B                           BIT     7,E
 0038'    CB 7C                           BIT     7,H
 003A'    CB 7D                           BIT     7,L
 003C'    CB 7E                           BIT     7,(HL)
 003E'    CB 7F                           BIT     7,A
 0040'    CB 80                           RES     0,B
 0042'    CB 81                           RES     0,C
 0044'    CB 82                           RES     0,D
 0046'    CB 83                           RES     0,E
 0048'    CB 84                           RES     0,H
 004A'    CB 85                           RES     0,L
 004C'    CB 86                           RES     0,(HL)
 004E'    CB 87                           RES     0,A
                                          END
```

```
        MACRO-80 3.3    09-MAY-79       PAGE    S
MACROS.
SYMBOLS.
NO  FATAL ERROR(S)

      MACRO-80 3.3    09-MAY-79       PAGE    1
                              00100   .Z80
                              00200   ;   NUMERIC LIST OF INSTRUCTION SET.6
                              00300   ;
0000'   CB 88                 00400          RES     1,B
0002'   CB 89                 00500          RES     1,C
0004'   CB 8A                 00600          RES     1,D
0006'   CB 8B                 00700          RES     1,E
0008'   CB 8C                 00800          RES     1,H
000A'   CB 8D                 00900          RES     1,L
000C'   CB 8E                 01000          RES     1,(HL)
000E'   CB 8F                 01100          RES     1,A
0010'   CB 90                 01200          RES     2,B
0012'   CB 91                 01300          RES     2,C
0014'   CB 92                 01400          RES     2,D
0016'   CB 93                 01500          RES     2,E
0018'   CB 94                 01600          RES     2,H
001A'   CB 95                 01700          RES     2,L
001C'   CB 96                 01800          RES     2,(HL)
001E'   CB 97                 01900          RES     2,A
0020'   CB 98                 02000          RES     3,B
0022'   CB 99                 02100          RES     3,C
0024'   CB 9A                 02200          RES     3,D
0026'   CB 9B                 02300          RES     3,E
0028'   CB 9C                 02400          RES     3,H
002A'   CB 9D                 02500          RES     3,L
002C'   CB 9E                 02600          RES     3,(HL)
002E'   CB 9F                 02700          RES     3,A
0030'   CB A0                 02800          RES     4,B
0032'   CB A1                 02900          RES     4,C
0034'   CB A2                 03000          RES     4,D
0036'   CB A3                 03100          RES     4,E
0038'   CB A4                 03200          RES     4,H
003A'   CB A5                 03300          RES     4,L
003C'   CB A6                 03400          RES     4,(HL)
003E'   CB A7                 03500          RES     4,A
0040'   CB A8                 03600          RES     5,B
0042'   CB A9                 03700          RES     5,C
0044'   CB AA                 03800          RES     5,D
0046'   CB AB                 03900          RES     5,E
0048'   CB AC                 04000          RES     5,H
004A'   CB AD                 04100          RES     5,L
004C'   CB AE                 04200          RES     5,(HL)
004E'   CB AF                 04300          RES     5,A
                              04400          END
```

```
        MACRO-80 3.3    09-MAY-79       PAGE    S
MACROS.
SYMBOLS:
NO  FATAL ERROR(S)

       MACRO-80 3.3    09-MAY-79       PAGE    1
                                00100   .Z80
                                00200   ;   NUMERIC LIST OF INSTRUCTION SET.7
                                00300   ;
 0000'    CB B0                 00400        RES     6,B
 0002'    CB B1                 00500        RES     6,C
 0004'    CB B2                 00600        RES     6,D
 0006'    CB B3                 00700        RES     6,E
 0008'    CB B4                 00800        RES     6,H
 000A'    CB B5                 00900        RES     6,L
 000C'    CB B6                 01000        RES     6,(HL)
 000E'    CB B7                 01100        RES     6,A
 0010'    CB B8                 01200        RES     7,B
 0012'    CB B9                 01300        RES     7,C
 0014'    CB BA                 01400        RES     7,D
 0016'    CB BB                 01500        RES     7,E
 0018'    CB BC                 01600        RES     7,H
 001A'    CB BD                 01700        RES     7,L
 001C'    CB BE                 01800        RES     7,(HL)
 001E'    CB BF                 01900        RES     7,A
 0020'    CB C0                 02000        SET     0,B
 0022'    CB C1                 02100        SET     0,C
 0024'    CB C2                 02200        SET     0,D
 0026'    CB C3                 02300        SET     0,E
 0028'    CB C4                 02400        SET     0,H
 002A'    CB C5                 02500        SET     0,L
 002C'    CB C6                 02600        SET     0,(HL)
 002E'    CB C7                 02700        SET     0,A
 0030'    CB C8                 02800        SET     1,B
 0032'    CB C9                 02900        SET     1,C
 0034'    CB CA                 03000        SET     1,D
 0036'    CB CB                 03100        SET     1,E
 0038'    CB CC                 03200        SET     1,H
 003A'    CB CD                 03300        SET     1,L
 003C'    CB CE                 03400        SET     1,(HL)
 003E'    CB CF                 03500        SET     1,A
 0040'    CB D0                 03600        SET     2,B
 0042'    CB D1                 03700        SET     2,C
 0044'    CB D2                 03800        SET     2,D
 0046'    CB D3                 03900        SET     2,E
 0048'    CB D4                 04000        SET     2,H
 004A'    CB D5                 04100        SET     2,L
 004C'    CB D6                 04200        SET     2,(HL)
 004E'    CB D7                 04300        SET     2,A
                                04400        END
```

```
      MACRO-80 3.3    09-MAY-79      PAGE    5
MACROS.
SYMBOLS.
NO  FATAL ERROR(S)

     MACRO-80 3.3    09-MAY-79      PAGE    1
                               00100   .Z80
                               00200   ;    NUMERIC LIST OF INSTRUCTION SET.8
                               00300   ;
0000'   CB D8                  00400           SET     3,B
0002'   CB D9                  00500           SET     3,C
0004'   CB DA                  00600           SET     3,D
0006'   CB DB                  00700           SET     3,E
0008'   CB DC                  00800           SET     3,H
000A'   CB DD                  00900           SET     3,L
000C'   CB DE                  01000           SET     3,(HL)
000E'   CB DF                  01100           SET     3,A
0010'   CB E0                  01200           SET     4,B
0012'   CB E1                  01300           SET     4,C
0014'   CB E2                  01400           SET     4,D
0016'   CB E3                  01500           SET     4,E
0018'   CB E4                  01600           SET     4,H
001A'   CB E5                  01700           SET     4,L
001C'   CB E6                  01800           SET     4,(HL)
001E'   CB E7                  01900           SET     4,A
0020'   CB E8                  02000           SET     5,B
0022'   CB E9                  02100           SET     5,C
0024'   CB EA                  02200           SET     5,D
0026'   CB EB                  02300           SET     5,E
0028'   CB EC                  02400           SET     5,H
002A'   CB ED                  02500           SET     5,L
002C'   CB EE                  02600           SET     5,(HL)
002E'   CB EF                  02700           SET     5,A
0030'   CB F0                  02800           SET     6,B
0032'   CB F1                  02900           SET     6,C
0034'   CB F2                  03000           SET     6,D
0036'   CB F3                  03100           SET     6,E
0038'   CB F4                  03200           SET     6,H
003A'   CB F5                  03300           SET     6,L
003C'   CB F6                  03400           SET     6,(HL)
003E'   CB F7                  03500           SET     6,A
0040'   CB F8                  03600           SET     7,B
0042'   CB F9                  03700           SET     7,C
0044'   CB FA                  03800           SET     7,D
0046'   CB FB                  03900           SET     7,E
0048'   CB FC                  04000           SET     7,H
004A'   CB FD                  04100           SET     7,L
004C'   CB FE                  04200           SET     7,(HL)
004E'   CB FF                  04300           SET     7,A
                               04400           END
```

```
          MACRO-80 3.3    09-MAY-79       PAGE    S
MACROS:
SYMBOLS:
NO  FATAL ERROR(S)




        MACRO-80 3.3    09-MAY-79       PAGE    1
                                00100   .Z80
                                00200   ;   NUMERIC LIST OF INSTRUCTION SET.9
                                00300
0012                            00400   X           EQU     012H
0034                            00500   Y           EQU     034H
1234                            00600   XY          EQU     01234H
0000'   CC 1234                 00700               CALL    Z,XY
0003'   CD 1234                 00800               CALL    XY
0006'   CE 12                   00900               ADC     A,X
0008'   CF                      01000               RST     8
0009'   D0                      01100               RET     NC
000A'   D1                      01200               POP     DE
000B'   D2 1234                 01300               JP      NC,XY
000E'   D3 12                   01400               OUT     (X),A
0010'   D4 1234                 01500               CALL    NC,XY
0013'   D5                      01600               PUSH    DE
0014'   D6 12                   01700               SUB     X
0016'   D7                      01800               RST     10H
0017'   D8                      01900               RET     C
0018'   D9                      02000               EXX
0019'   DA 1234                 02100               JP      C,XY
001C'   DB 12                   02200               IN      A,(X)
001E'   DC 1234                 02300               CALL    C,XY
0021'   DD 09                   02400               ADD     IX,BC
0023'   DD 19                   02500               ADD     IX,DE
0025'   DD 21 1234              02600               LD      IX,XY
0029'   DD 22 1234              02700               LD      (XY),IX
002D'   DD 23                   02800               INC     IX
002F'   DD 29                   02900               ADD     IX,IX
0031'   DD 2A 1234              03000               LD      IX,(XY)
0035'   DD 2B                   03100               DEC     IX
0037'   DD 34 12                03200               INC     (IX+X)
003A'   DD 35 12                03300               DEC     (IX+X)
003D'   DD 36 12 34             03400               LD      (IX+X),Y
0041'   DD 39                   03500               ADD     IX,SP
0043'   DD 46 12                03600               LD      B,(IX+X)
0046'   DD 4E 12                03700               LD      C,(IX+X)
0049'   DD 56 12                03800               LD      D,(IX+X)
004C'   DD 5E 12                03900               LD      E,(IX+X)
004F'   DD 66 12                04000               LD      H,(IX+X)
0052'   DD 6E 12                04100               LD      L,(IX+X)
                                04200               END
```

```
        MACRO-80 3.3    09-MAY-79         PAGE   S
MACROS:
SYMBOLS:
X         0012    XY       1234  Y         0034
NO  FATAL ERROR(S)

      MACRO-80 3.3   09-MAY-79         PAGE    1
                                 00100   .Z80
                                 00200   ;   NUMERIC LIST OF INSTRUCTION SET.10
                                 00300   ;
0012                             00400   X       EQU     012H
0034                             00500   Y       EQU     034H
1234                             00600   XY      EQU     01234H
0000'   DD 70 12                 00700           LD      (IX+X),B
0003'   DD 71 12                 00800           LD      (IX+X),C
0006'   DD 72 12                 00900           LD      (IX+X),D
0009'   DD 73 12                 01000           LD      (IX+X),E
000C'   DD 74 12                 01100           LD      (IX+X),H
000F'   DD 75 12                 01200           LD      (IX+X),L
0012'   DD 77 12                 01300           LD      (IX+X),A
0015'   DD 7E 12                 01400           LD      A,(IX+X)
0018'   DD 86 12                 01500           ADD     A,(IX+X)
001B'   DD 8E 12                 01600           ADC     A,(IX+X)
001E'   DD 96 12                 01700           SUB     (IX+X)
0021'   DD 9E 12                 01800           SBC     A,(IX+X)
0024'   DD A6 12                 01900           AND     (IX+X)
0027'   DD AE 12                 02000           XOR     (IX+X)
002A'   DD B6 12                 02100           OR      (IX+X)
002D'   DD BE 12                 02200           CP      (IX+X)
0030'   DD E1                    02300           POP     IX
0032'   DD E3                    02400           EX      (SP),IX
0034'   DD E5                    02500           PUSH    IX
0036'   DD E9                    02600           JP      (IX)
0038'   DD F9                    02700           LD      SP,IX
003A'   DD CB 12 06              02800           RLC     (IX+X)
003E'   DD CB 12 0E              02900           RRC     (IX+X)
0042'   DD CB 12 16              03000           RL      (IX+X)
0046'   DD CB 12 1E              03100           RR      (IX+X)
004A'   DD CB 12 26              03200           SLA     (IX+X)
004E'   DD CB 12 2E              03300           SRA     (IX+X)
0052'   DD CB 12 3E              03400           SRL     (IX+X)
0056'   DD CB 12 46              03500           BIT     0,(IX+X)
005A'   DD CB 12 4E              03600           BIT     1,(IX+X)
005E'   DD CB 12 56              03700           BIT     2,(IX+X)
0062'   DD CB 12 5E              03800           BIT     3,(IX+X)
0066'   DD CB 12 66              03900           BIT     4,(IX+X)
006A'   DD CB 12 6E              04000           BIT     5,(IX+X)
006E'   DD CB 12 76              04100           BIT     6,(IX+X)
0072'   DD CB 12 7E              04200           BIT     7,(IX+X)
                                 04300           END
```

```
          MACRO-80 3.3    09-MAY-79        PAGE    5
MACROS.
SYMBOLS.
X        0012    XY       1234    Y        0034
NO  FATAL ERROR(S)

       MACRO-80 3.3    09-MAY-79        PAGE    1
                                   00100   .Z80
                                   00200   ;    NUMERIC LIST OF INSTRUCTION SET.11
                                   00300   ;
0012                               00400   X         EQU      012H
0034                               00500   Y         EQU      034H
1234                               00600   XY        EQU      01234H
0000'   DD CB 12 86                00700             RES      0,(IX+X)
0004'   DD CB 12 8E                00800             RES      1,(IX+X)
0008'   DD CB 12 96                00900             RES      2,(IX+X)
000C'   DD CB 12 9E                01000             RES      3,(IX+X)
0010'   DD CB 12 A6                01100             RES      4,(IX+X)
0014'   DD CB 12 AE                01200             RES      5,(IX+X)
0018'   DD CB 12 B6                01300             RES      6,(IX+X)
001C'   DD CB 12 BE                01400             RES      7,(IX+X)
0020'   DD CB 12 C6                01410             SET      0,(IX+X)
0024'   DD CB 12 CE                01420             SET      1,(IX+X)
0028'   DD CB 12 D6                01430             SET      2,(IX+X)
002C'   DD CB 12 DE                01440             SET      3,(IX+X)
0030'   DD CB 12 E6                01450             SET      4,(IX+X)
0034'   DD CB 12 EE                01460             SET      5,(IX+X)
0038'   DD CB 12 F6                01470             SET      6,(IX+X)
003C'   DD CB 12 FE                01480             SET      7,(IX+X)
0040'   DE 12                      01500             SBC      A,X
0042'   DF                         01600             RST      18H
0043'   E0                         01700             RET      PO
0044'   E1                         01800             POP      HL
0045'   E2 1234                    01900             JP       PO,XY
0048'   E3                         02000             EX       (SP),HL
0049'   E4 1234                    02100             CALL     PO,XY
004C'   E5                         02200             PUSH     HL
004D'   E6 12                      02300             AND      X
004F'   E7                         02400             RST      20H
0050'   E8                         02500             RET      PE
0051'   E9                         02600             JP       (HL)
0052'   EA 1234                    02700             JP       PE,XY
0055'   EB                         02800             EX       DE,HL
0056'   EC 1234                    02900             CALL     PE,XY
0059'   ED 40                      03000             IN       B,(C)
005B'   ED 41                      03100             OUT      (C),B
005D'   ED 42                      03200             SBC      HL,BC
005F'   ED 43 1234                 03300             LD       (XY),BC
0063'   ED 44                      03400             NEG
                                   03500             END
```

```
          MACRO-80 3.3     09-MAY-79        PAGE     S
      MACROS.
      SYMBOLS.
      X       0012     XY       1234     Y       0034
      NO  FATAL ERROR(S)
        MACRO-80 3.3    09-MAY-79        PAGE   1
                                    00100   .Z80
                                    00200   ;    NUMERIC LIST OF INSTRUCTION SET.12
                                    00300   ;
    0012                            00400   X         EQU     012H
    0034                            00500   Y         EQU     034H
    1234                            00600   XY        EQU     01234H
    0000'    ED 45                  00700             RETN
    0002'    ED 46                  00800             IM      0
    0004'    ED 47                  00900             LD      I,A
    0006'    ED 48                  01000             IN      C,(C)
    0008'    ED 49                  01100             OUT     (C),C
    000A'    ED 4A                  01200             ADC     HL,BC
    000C'    ED 4B 1234             01300             LD      BC,(XY)
    0010'    ED 4D                  01400             RETI
    0012'    ED 50                  01500             IN      D,(C)
    0014'    ED 51                  01600             OUT     (C),D
    0016'    ED 52                  01700             SBC     HL,DE
    0018'    ED 53 1234             01800             LD      (XY),DE
    001C'    ED 56                  01900             IM      1
    001E'    ED 57                  02000             LD      A,I
    0020'    ED 58                  02100             IN      E,(C)
    0022'    ED 58                  02200             IN      E,(C)
    0024'    ED 59                  02300             OUT     (C),E
    0026'    ED 5A                  02400             ADC     HL,DE
    0028'    ED 5B 1234             02500             LD      DE,(XY)
    002C'    ED 5E                  02600             IM      2
    002E'    ED 60                  03300             IN      H,(C)
    0030'    ED 61                  03400             OUT     (C),H
    0032'    ED 62                  03500             SBC     HL,HL
    0034'    ED 67                  03600             RRD
    0036'    ED 68                  03700             IN      L,(C)
    0038'    ED 69                  03800             OUT     (C),L
    003A'    ED 6A                  03900             ADC     HL,HL
    003C'    ED 6F                  04000             RLD
    003E'    ED 72                  04100             SBC     HL,SP
    0040'    ED 73 1234             04200             LD      (XY),SP
    0044'    ED 78                  04300             IN      A,(C)
    0046'    ED 79                  04400             OUT     (C),A,
    0048'    ED 7A                  04500             ADC     HL,SP
    004A'    ED 7B 1234             04600             LD      SP,(XY)
    004E'    ED A0                  04700             LDI
    0050'    ED A1                  04800             CPI
    0052'    ED A2                  04900             INI
    0054'    ED A3                  05000             OUTI
                                    05100             END
```

```
                          MACRO-80 3.3    09-MAY-79       PAGE . S
                 MACROS:
                 SYMBOLS:
                 X        0012    XY      1234    Y       0034
                 NO  FATAL ERROR(S)


          MACRO-80 3.3    09-MAY-79       PAGE    1
                                  00100   .Z80
                                  00200   ;      NUMERIC LIST OF INSTRUCTION SET.13
                                  00300   ;
  0012                            00400   X       EQU     012H
  0034                            00500   Y       EQU     034H
  1234                            00600   XY      EQU     01234H
  0000'   ED A8                   00700           LDD
  0002'   ED A9                   00800           CPD
  0004'   ED AA                   00900           IND
  0006'   ED AB                   01000           OUTD
  0008'   ED B0                   01100           LDIR
  000A'   ED B1                   01200           CPIR
  000C'   ED B2                   01300           INIR
  000E'   ED B3                   01400           OTIR
  0010'   ED B8                   01500           LDDR
  0012'   ED B9                   01600           CPDR
  0014'   ED BA                   01700           INDR
  0016'   ED BB                   01800           OTDR
  0018'   EE 12                   01900           XOR     X
  001A'   EF                      02000           RST     28H
  001B'   F0                      02100           RET     P
  001C'   F1                      02200           POP     AF
  001D'   F2 1234                 02300           JP      P,XY
  0020'   F3                      02400           DI
  0021'   F4 1234                 02500           CALL    P,XY
  0024'   F5                      02600           PUSH    AF
  0025'   F6 12                   02700           OR      X
  0027'   F7                      02800           RST     30H
  0028'   F8                      02900           RET     M
  0029'   F9                      03000           LD      SP,HL
  002A'   FA 1234                 03100           JP      M,XY
  002D'   FB                      03200           EI
  002E'   FC 1234                 03300           CALL    M,XY
  0031'   FD 09                   03500           ADD     IY,BC
  0033'   FD 19                   03600           ADD     IY,DE
  0035'   FD 21 1234              03700           LD      IY,XY
  0039'   FD 22 1234              03800           LD      (XY),IY
  003D'   FD 23                   03900           INC     IY
  003F'   FD 29                   04000           ADD     IY,IY
  0041'   DD 2A 1234              04100           LD      IY,(XY)
  0045'   FD 2B                   04200           DEC     IY
                                  04300           END
```

```
              MACRO-80 3.3    09-MAY-79       PAGE    S
        MACROS.
        SYMBOLS.
        X        0012    XY      1234    Y       0034
        NO  FATAL ERROR(S)
         MACRO-80 3.3    09-MAY-79       PAGE    1
                                    00100    Z80
                                    00200    ;   NUMERIC LIST OF INSTRUCTION SET.14
                                    00300    ;
   0012                             00400   X        EQU     012H
   0034                             00500   Y        EQU     034H
   1234                             00600   XY       EQU     01234H
   0000'   FD 34 12                 00700            INC     (IY+X)
   0003'   FD 35 12                 00800            DEC     (IY+X)
   0006'   FD 36 12 34              00900            LD      (IY+X),Y
   000A'   FD 39                    01000            ADD     IY,SP
   000C'   FD 46 12                 01100            LD      B,(IY+X)
   000F'   FD 4E 12                 01200            LD      C,(IY+X)
   0012'   FD 56 12                 01300            LD      D,(IY+X)
   0015'   FD 5E 12                 01400            LD      E,(IY+X)
   0018'   FD 66 12                 01500            LD      H,(IY+X)
   001B'   FD 6E 12                 01600            LD      L,(IY+X)
   001E'   FD 70 12                 01700            LD      (IY+X),B
   0021'   FD 71 12                 01800            LD      (IY+X),C
   0024'   FD 72 12                 01900            LD      (IY+X),D
   0027'   FD 73 12                 02000            LD      (IY+X),E
   002A'   FD 74 12                 02100            LD      (IY+X),H
   002D'   FD 75 12                 02200            LD      (IY+X),L
   0030'   FD 77 12                 02300            LD      (IY+X),A
   0033'   FD 7E 12                 02400            LD      A,(IY+X)
   0036'   FD 86 12                 02500            ADD     A,(IY+X)
   0039'   FD 8E 12                 02600            ADC     A,(IY+X)
   003C'   FD 96 12                 02700            SUB     (IY+X)
   003F'   FD 9E 12                 02800            SBC     A,(IY+X)
   0042'   FD A6 12                 02900            AND     (IY+X)
   0045'   FD AE 12                 03000            XOR     (IY+X)
   0048'   FD B6 12                 03100            OR      (IY+X)
   004B'   FD BE 12                 03200            CP      (IY+X)
   004E'   FD E1                    03300            POP     IY
   0050'   FD E3                    03400            EX      (SP),IY
   0052'   FD E5                    03500            PUSH    IY
   0054'   FD E9                    03600            JP      (IY)
   0056'   FD F9                    03700            LD      SP,IY
   0058'   FD CB 12 06              03800            RLC     (IY+X)
   005C'   FD CB 12 0E              03900            RRC     (IY+X)
   0060'   FD CB 12 16              04000            RL      (IY+X)
   0064'   FD CB 12 1E              04100            RR      (IY+X)
   0068'   FD CB 12 26              04200            SLA     (IY+X)
   006C'   FD CB 12 2E              04300            SRA     (IY+X)
   0070'   FD CB 12 3E              04400            SRL     (IY+X)
                                    04500            END
```

```
                MACRO-80 3.3    09-MAY-79        PAGE    S
        MACROS.
        SYMBOLS.
        X       0012    XY      1234    Y        0034
        NO  FATAL ERROR(S)


MACRO-80 3.3    09-MAY-79       PAGE    1
                        00100   .Z80
                        00200   ;   NUMERIC LIST OF INSTRUCTION SET.15
                        00300   ;
0012                    00400   X       EQU     012H
0000'   FD CB 12 46     00500           BIT     0,(IY+X)
0004'   FD CB 12 4E     00600           BIT     1,(IY+X)
0008'   FD CB 12 56     00700           BIT     2,(IY+X)
000C'   FD CB 12 5E     00800           BIT     3,(IY+X)
0010'   FD CB 12 66     00900           BIT     4,(IY+X)
0014'   FD CB 12 6E     01000           BIT     5,(IY+X)
0018'   FD CB 12 76     01100           BIT     6,(IY+X)
001C'   FD CB 12 7E     01200           BIT     7,(IY+X)
0020'   FD CB 12 86     01300           RES     0,(IY+X)
0024'   FD CB 12 8E     01400           RES     1,(IY+X)
0028'   FD CB 12 96     01500           RES     2,(IY+X)
002C'   FD CB 12 9E     01600           RES     3,(IY+X)
0030'   FD CB 12 A6     01700           RES     4,(IY+X)
0034'   FD CB 12 AE     01800           RES     5,(IY+X)
0038'   FD CB 12 B6     01900           RES     6,(IY+X)
003C'   FD CB 12 BE     02000           RES     7,(IY+X)
0040'   FD CB 12 C6     02100           SET     0,(IY+X)
0044'   FD CB 12 CE     02200           SET     1,(IY+X)
0048'   FD CB 12 D6     02300           SET     2,(IY+X)
004C'   FD CB 12 DE     02400           SET     3,(IY+X)
0050'   FD CB 12 E6     02500           SET     4,(IY+X)
0054'   FD CB 12 EE     02600           SET     5,(IY+X)
0058'   FD CB 12 F6     02700           SET     6,(IY+X)
005C'   FD CB 12 FE     02800           SET     7,(IY+X)
0060'   FE 12           02900           CP      X
0062'   FF              03000           RST     38H
                        03100           END


                MACRO-80 3.3    09-MAY-79        PAGE    S
        MACROS:
        SYMBOLS:
        X       0012
        NO  FATAL ERROR(S)
```

システム・コンポーネント・マイクロコンピュータ

Logitec LDS-II



Z-80搭載!!

メモリ空間は4Mバイトまで

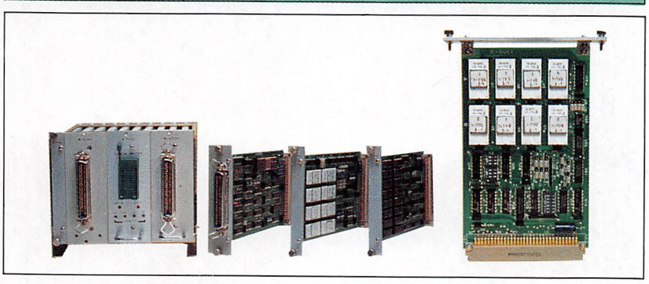
FDOSはCP/M® を採用。

《主な仕様》

◎ CPUは、Z-80を使用
◎ RS232C　シリアルポート、2ポート
◎ フロッピーディスクドライブ2台(FDCは、4台迄可)
◎ RAM32K：64K迄増設可能
◎ ROM2K：32K迄増設可能
◎ 6ビットのセグメント指定可能
◎ シリアルI/oポートにより、CRTターミナル、プリンタ
　プロッタ、ユーザーシステム等のI/o機器との接続が可能。
◎ BASIC、FORTRAN等の高級言語使用可能

注)、CP／Mは米国デジタルリサーチ社の登録商標です。

K-8000ファミリーで、強力なサポート。



LDS-IIを、サポートするK-8000シリーズ(システムコンポーネントモジュール)の各種ファミリーは、マイコン応用のシステム機器の開発が、短期間でしかも経済的にできます。K-8000シリーズは、内外の最新の情報と技術を導入しており、ボードサイズは、100×160㎜(ヨーロッパフォーマット、EIA規格の標準ラックにコンパチブル)と、小型でありながら多層基板の採用により各モジュールの機能は、アップされており特にノイズ等には、万全の注意がなされております。

関東電子機器販売株式会社

システム販売部システム課	☎ 03 (251) 2 9 2 1
大阪支店	☎ 06 (632) 0207～9
群馬営業所	☎ 0270 (23) 2 3 0 1
福岡営業所	☎ 092 (713) 1 2 9 9
名古屋営業所	☎ 052 (263) 1 6 2 9

関東 Byteショップ	☎03 (253) 5 2 6 4
名古屋 Byteショップ	☎052 (263) 1 6 2 9
大阪 Byteショップ	☎06 (644) 1 5 4 8
福岡 Byteショップ	☎092 (713) 1 2 9 8
岡谷 Byteショップ	☎02662 (3) 1 0 7 5
伊勢崎 Byteショップ	☎0270 (23) 2 3 0 2
秋葉原 Byteショップ KOYO	☎03 (255) 6 5 0 4

Z80マイコンプログラムテクニック 昭和五十五年四月三十日発行 電波新聞社 〒141 東京都品川区東五反田一―十一―十五 （〇三）四四五―六一一一（大代）

電波新聞社 〒141東京都品川区東五反田1－11－15電話03（445）6111 定価1300円 雑誌08370－4